```
#include <iostream>
#include <fstream>
#include <conio.h>
#include <string>
#include <windows.h>
#include <iomanip>
using namespace std;
struct Detail {
    string name;
    string authur;
    double price;
    int quantity;
};
int length = 1, Index = 0;
Detail Books[1000];
Detail SoldBooks[1000];
int SaleIndex = 0, SaleLength = 1;
int BookFound1 = 0;
int BookFound = 0;
int choice = 0;
int Design = 0;
ofstream File;
ifstream Infile;
void SaleBook();
void ViewSoldBook();
void SaveSoldData();
void ReadSoldData();
void MainFunction();
int main();
void SaveData() {
    File.open("TestBook.J", ios::app);
    for (int i = 0; i < Books[Index].name.length();
        i++) {
        if (Books[Index].name[i] == 32) {
            Books[Index].name[i] = 95;
        }
    }
    for (int i = 0; i < Books[Index].authur.length();
        i++) {
        if (Books[Index].authur[i] == 32) {
            Books[Index].authur[i] = 95;
        }
    }
    for (int i = length - 1; i < length; i++) {
        File << " " << Books[Index].name << endl
            << Books[Index].authur << endl
            << Books[Index].price << endl
            << Books[Index].quantity << endl
            << endl;
    }
    File.close();
}
void ReadData() {
    Infile.open("TestBook.J");
    while (!Infile.eof()) {

        Infile >> Books[Index].name;
        Infile >> Books[Index].authur;
        Infile >> Books[Index].price;
        Infile >> Books[Index].quantity;
        if (!Infile.eof()) {

            Index = Index + 1;
            length++;
        }
    }
    Infile.close();
}
bool IsValid(int Num) {
    return ((Num >= 1) && (Num <= 9));
}
int SearchBook1() {
retry:
    system("cls");
    cout << endl << endl << endl;
    cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2     "
        "Book  Search      "
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\n";
    cout << endl;
    cout << "\t\tEnter Book name: ";
    string BookSearch;
    cin.ignore();
    getline(cin, BookSearch);
    for (int i = 0; i < BookSearch.length(); i++) {
        if (BookSearch[i] == 32) {
            BookSearch[i] = 95;
        }
    }

    for (int i = 0; i < length; i++) {
        if (BookSearch == Books[i].name) {
            BookFound = 1;
            BookFound1 = 1;
            int Index = i;
            return Index;
        }

    }
}
int SearchBook() {
retry:
    system("cls");
    cout << endl << endl << endl;
    cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2     "
        "Book  Search      "
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\n";
    cout << endl;
    cout << "\t\tEnter Book name: " << endl;
    string BookSearch;

    getline(cin, BookSearch);
  " || "(int i = 0; i < BookSearch.length(); i++
        if (BookSearch[i] == 32) {
            BookSearch[i] = 95;
        }
    }
    for (int i = 0; i < SaleLength; i++) {
        if (BookSearch == SoldBooks[i].name) {
        retryS:
            cout << "\t\tBook Found: ";
            cout << "How Much Custumer Want To Buy: ";
            int tempQ;
            cin >> tempQ;
            if (cin.fail()) {
                cin.ignore();
                cin.clear();
                cout << "Enter A Valid Quantity: ";
                Sleep(500);
                goto retryS;
            }
            SoldBooks[i].quantity = SoldBooks[i].quantity + tempQ;
            File.open("Salerecord.J");
            for (int i = 0; i < SaleIndex; i++) {
                File << " " << SoldBooks[i].name << endl
                    << SoldBooks[i].authur << endl
                    << SoldBooks[i].price << endl
                    << SoldBooks[i].quantity << endl
                    << endl;
            }
            File.close();
            int Pricee;
            for (int k = 0; k < length; k++) {
                if (SoldBooks[i].name == Books[k].name) {
                    Books[k].quantity = Books[k].quantity - tempQ;
                    cout << "The Total Bill Is: " << Books[k].price * tempQ << endl;
                    break;
                }
            }

            File.open("TestBook.J");
            for (int i = 0; i < Index; i++) {
                File << " " << Books[i].name << endl
                    << Books[i].authur << endl
                    << Books[i].price << endl
                    << Books[i].quantity << endl
                    << endl;
            }
            File.close();
            system("pause");
            system("cls");
            MainFunction();
        }
    }
    for (int i = 0; i < length; i++) {

        if (BookSearch == Books[i].name) {
            BookFound = 1;
            BookFound1 = 1;
            int Index = i;
            return Index;
        }
    }
}
int AuthurBookCount(string Authur) {
    int AuthurCount = 0;
    for (int i = 0; i < length; i++) {
        if (Authur == Books[i].authur) {
            AuthurCount++;
        }
    }
    return AuthurCount;
}
void AuthurShowBooks(string Authur) {
    system("cls");
    cout << endl << endl << endl;
    cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2     "
        "Book  Search      "
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\n";
    cout << endl;
    cout << "\t\tBooks By The Same Authur" << endl;
    for (int i = 0; i < length; i++) {
        if (Authur == Books[i].authur) {
            cout << "***************************************************************************"
```

```
                << endl;
            cout << "Book Name: " << Books[i].name << " || "
                << " Authur Name: " << Books[i].authur << " || "
                << " Book Price: " << Books[i].price << " || "
                << " Quantity: " << Books[i].quantity << endl;
            cout << "**************************************************************************"
                << endl;
        }
    }
}
void SearchAuthur() {
    system("cls");
    cout << endl << endl << endl;
    cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2     "
        "Book  Search      "
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
        "\n";
    cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\n";
    cout << endl;
    cout << "\t\tEnter Auther's name: ";
    int Authurcheck = 0;
    string AuthurSearch;

    cin.ignore();
    getline(cin, AuthurSearch);
    for (int i = 0; i < AuthurSearch.length(); i++) {
        if (AuthurSearch[i] == 32) {
            AuthurSearch[i] = 95;
        }
    }
    for (int i = 0; i < length; i++) {
        if (AuthurSearch == Books[i].authur) {
            cout << "**************************************************************************"
                << endl;
            cout << "Book Name: " << Books[i].name << " || "
                << " Authur Name: " << Books[i].authur << " || "
                << " Book Price: " << Books[i].price << " || "
  " || "        << " Quantity: " <<
            cout << "**************************************************************************"
                << endl;
            Authurcheck++;
        }
    }
    if (Authurcheck != 0) {
        cout << Authurcheck <<
            " Books Found By the Authur's Name: " << endl;
```

```cpp
    }

    if (Authurcheck == 0) {
        cout << "No Book Found Under The Given Authur name: "
            << endl;
    }
    system("pause");
}
void MainFunction(){
    while (choice != 8) {
    retry:
        system("cls");
        cout << endl << endl << endl;
        cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
            "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
            "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
            "\xdb\xdb\xdb\n";
        cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
            "\xdb\xdb\xdb\n";
        cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2 "
            "Book Sales management "
            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb"
            "\xdb\n";
        cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
            "\xdb\xdb\xdb\n";
        cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
            "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
            "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
            "\xdb\xdb\xdb\n";
        cout << endl;
        cout << "\t\t1. Add Book" << endl;
        cout << "\t\t2. Delete Book" << endl;
        cout << "\t\t3. Update Book" << endl;
        cout << "\t\t4. View All books" << endl;
        cout << "\t\t5. Search Book" << endl;
        cout << "\t\t6. Sell a Book" << endl;
        cout << "\t\t7. View Sold Book" << endl;
        \
            cout << "\t\t8. Exit Program" << endl;
        cin >> choice;
        if (cin.fail()) {
            cin.clear();
            cin.ignore();
            goto retry;
        }
        if (!IsValid(choice)) {
            goto retry;
        }
        cin.ignore();
        if (choice > 0 && choice <= 8) {

            if (choice == 1) {
                int AuthurTaker = 1, lengthIn = 0;

                system("cls");
                cout << endl << endl << endl;
                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
            cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                "\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
            cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                "\xb2     Book  Addition    "
                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                "\xdb\xdb\xdb\n";
            cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                "\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
            cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
            cout << endl;
            for (int i = length - 1; i < length; i++) {

                cout << "\t\tbook Name:          ";
                getline(cin, Books[Index].name);
                for (int i = 0; i < Books[Index].name.length();
                    i++) {
                    if (Books[Index].name[i] == 32) {
                        Books[Index].name[i] = 95;
                    }
                }
                for (int i = Index - 1; i >= 0; i--) {
                    if (Books[Index].name == Books[i].name) {
                        AuthurTaker = 0;
                        lengthIn = 0;
                        cout << "Book Already Exist: " << endl;
                        cout << "Do You Want to Add Book Quantity:  y/n"
                            << endl;
                        char Temp_Choice;
                        cin >> Temp_Choice;
                        if (Temp_Choice == 'y') {
                            system("cls");
                            Design = 1;
                            if (Design) {
                                cout << endl << endl << endl;
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2     Book  Addition    "
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << endl;
                                Design = 0;
                            }
                        RetryTemp_Quantity:
                            if (Design) {
                                cout << endl << endl << endl;
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2     Book  Addition    "
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << endl;
                                Design = 0;
                            }
                            int Temp_Quantity;
                            cout << "\t\tCurrent Quantity of Books is: " <<
                                Books[i].quantity
                                << endl;
                            cout << "\t\tHow Much You Want to Add: ";
                            cin >> Temp_Quantity;
                            if (cin.fail()) {
                                cin.clear();
                                cin.ignore();
                                cout << "Please Enter A Valid Input this time: "
                                    << endl;
                                Sleep(1000);
                                system("cls");
                                Design = 1;
                                goto RetryTemp_Quantity;
                            }
                            cin.ignore();
                            Books[i].quantity = Books[i].quantity +
                                Temp_Quantity;
                            cout << "The New Quantity is: " <<
                                Books[i].quantity << endl;
```

```
" || "
                          }
                          File.open("TestBook.J");
                          for (int i = 0; i < Index; i++) {
                              File << " " << Books[i].name << endl
                                  << Books[i].authur << endl
                                  << Books[i].price << endl
                                  << Books[i].quantity << endl
                                  << endl;
                          }
                          File.close();
                      }
                  }
                  if (AuthurTaker) {
                      lengthIn = 1;
                      cout << "\t\tAuthur Name:        ";
                      getline(cin, Books[Index].authur);
                      for (int i = 0; i < Books[Index].authur.length();
                          i++) {
                          if (Books[Index].authur[i] == 32) {
                              Books[Index].authur[i] = 95;
                          }
                      }
                  RetryPrice:
                      if (Design) {
                          cout << endl << endl << endl;
                          cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                              "\xdb\xdb\xdb\n";
                          cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                              "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                              "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                              "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                              "\xdb\xdb\xdb\n";
                          cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                              "\xb2\xb2\xb2\xb2     Book  Addition    "
                              "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                              "\xb2\xb2\xb2\xdb\xdb\xdb\n";
                          cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                              "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                              "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                              "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                              "\xdb\xdb\xdb\n";
                          cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                              "\xdb\xdb\xdb\n";
                          cout << endl;
                          Design = 0;
                      }
                      cout << "\t\tPrice of the Book:    ";
                      cin >> Books[Index].price;
                      if (cin.fail()) {
                          cin.clear();
                          cin.ignore();
                          cout << "Please Enter A Valid Input this time: "
                              << endl;
                          Sleep(1000);
```

```
                            Design = 1;
                            system("cls");
                            goto RetryPrice;
                        }
                        cin.ignore();
                    RetryQuantity:
                        if (Design) {
                            cout << endl << endl << endl;
                            cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\n";
                            cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xdb\xdb\xdb\n";
                            cout << "\t\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2     Book  Addition    "
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xdb\xdb\xdb\n";
                            cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xdb\xdb\xdb\n";
                            cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\n";
                            cout << endl;
                            Design = 0;
                        }
                        cout << "\t\tQuantity of the Book: ";
                        cin >> Books[Index].quantity;
                        if (cin.fail()) {
                            cin.clear();
                            cin.ignore();
                            cout << "Please Enter A Valid Input this time: "
                                << endl;
                            Sleep(1000);
                            system("cls");
                            Design = 1;
                            goto RetryQuantity;
                        }
                        cin.ignore();
                        SaveData();
                        cout << "\t\tThe book Is Added: " << endl;
                        system("pause");
                    }
                }
                if (lengthIn) {
                    length++;
                    Index++;
                }
            }
            else if (choice == 2) {
                string Delete;
                system("cls");
```

```cpp
                cout << endl << endl << endl;
                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2      Delete Book      "
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xdb\xdb\xdb\n";
                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                cout << endl;
                cout << "\t\tEnter The Book Name: ";
                getline(cin, Delete);
                for (int i = 0; i < Delete.length(); i++) {
                    if (Delete[i] == 32) {
                        Delete[i] = 95;
                    }
                }
                int BookFoundChecker = 0;
                for (int i = 0; i < length; i++) {
                    if (Delete == Books[i].name) {
                        cout << "Book Found: " << endl;
                    RetryDeleteDecision:
                        if (Design) {
                            cout << endl << endl << endl;
                            cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\n";
                            cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xdb\xdb\xdb\n";
                            cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2      Delete Book      "
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xdb\xdb\xdb\n";
                            cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                "\xdb\xdb\xdb\n";
                            cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                "\xdb\xdb\xdb\n";
                            cout << endl;
                            Design = 0;
                        }
                        cout << "\t\t1. Delete Entire Book From System: "
                            << endl;
                        cout << "\t\t2. Reduce The Quanity Of Book: " <<
                            endl;
                        int DeleteChoice;
                        cin >> DeleteChoice;
                        if (cin.fail()) {
                            cin.clear();
                            cin.ignore();
                            cout << "Please Enter A Valid Input: ";
                            Sleep(250);
                            system("cls");
                            Design = 1;
                            goto RetryDeleteDecision;
                        }
                        if (DeleteChoice == 1) {
                            system("cls");
                            Design = 1;
                            if (Design) {
                                cout << endl << endl << endl;
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2      Delete Book      "
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << endl;
                                Design = 0;
                            }
                            for (int k = i; k < length; k++) {
                                if (k + 1 <= length - 1) {
                                    string temp = Books[k].name;
                                    Books[k].name = Books[k + 1].name;
                                    Books[k + 1].name = temp;
                                }
                                if (k + 1 <= length - 1) {
                                    string temp = Books[k].authur;
                                    Books[k].authur = Books[k + 1].authur;
                                    Books[k + 1].authur = temp;
                                }
                                if (k + 1 <= length - 1) {
                                    float temp = Books[k].price;
                                    Books[k].price = Books[k + 1].price;
                                    Books[k + 1].price = temp;
                                }
                                if (k + 1 <= length - 1) {
                                    int temp = Books[k].quantity;
                                    Books[k].quantity = Books[k + 1].quantity;
                                    Books[k + 1].quantity = temp;
                                }
                            }
                            length = length - 1;
                            Index = Index - 1;
                            BookFoundChecker++;
                            cout << "Book Deleted Successfully: " << endl;

                            Sleep(500);
                        }
                        else if (DeleteChoice == 2) {
                            system("cls");
                            Design = 1;
                            if (Design) {
                                cout << endl << endl << endl;
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2      Delete Book      "
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << endl;
                                Design = 0;
                            }
                            cout << "\t\tThe Current Quantity Of the Book Named: "
                                << Books[i].name
                                << " is: " << Books[i].quantity << endl;
                        RetryTempDel:
                            if (Design) {
                                cout << endl << endl << endl;
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
```

```
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2      Delete Book      "
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << endl;
                                Design = 0;
                            }
                            cout << "\t\tHow Much You Want To Reduce: ";
                            int TempDelQuantity;
                            cin >> TempDelQuantity;
                            if (cin.fail()) {
                                cin.clear();
                                cin.ignore();
                                cout << "Enter A valid Input: ";
                                Sleep(250);
                                system("cls");
                                Design = 1;
                                goto RetryTempDel;
                            }
                            Books[i].quantity = Books[i].quantity -
                                TempDelQuantity;
                            cout << "Quantity Updated: " << endl;
                            cout << "The New Quantity is: " <<
                                Books[i].quantity << endl;
                            Sleep(300);
                        }
                        else {
                            cout << "Please Enter From Given Number: " <<
                                endl;
                            Sleep(250);
                            system("cls");
                            Design = 1;
                            goto RetryDeleteDecision;
                        }
                    }
                }
                if (BookFoundChecker == 0) {
                    cout << "Book Not Found: ";
                }
                File.open("TestBook.J");
                for (int i = 0; i < Index; i++) {
                    File << " " << Books[i].name << endl
                        << Books[i].authur << endl
                        << Books[i].price << endl
                        << Books[i].quantity << endl
                        << endl;
                }
                File.close();
            }

            else if (choice == 3) {
                system("cls");
                int BookIndex = SearchBook();
                if (BookFound1) {
                    cout << "Book Found: " << endl;
                    char LoopChoice;
                    do {
                    RetryDecision:
                        system("cls");
                        cout << endl << endl << endl;
                        cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                            "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                            "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                            "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                        cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                        cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                            "\xb2\xb2\xb2 Book Sales management "
                            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                            "\xb2\xb2\xdb\xdb\xdb\n";
                        cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                            "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                        cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                            "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                            "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                            "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                        cout << endl;
                        cout << "\t\t1. Update Price: " << endl;
                        cout << "\t\t2. Update Quantity: " << endl;
                        int UpdateDecision;
                        cin >> UpdateDecision;
                        if (cin.fail()) {
                            cin.clear();
                            cin.ignore();
                            cout << "Please Enter A Valid Input this time: "
                                << endl;
                            Sleep(1000);
                            goto RetryDecision;
                        }
                        if (UpdateDecision == 1 || UpdateDecision == 2) {
                            cin.ignore();
                            BookFound = 0;
                            switch (UpdateDecision) {
                            case 1: {
                            RetryTempPrice:
                                system("cls");
                                cout << endl << endl << endl;
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2    Price Updating     "
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << endl;
                                cout << "\t\tThe Old Price Is :" <<
                                    Books[BookIndex].price
                                    << endl;
                                cout << "\t\tEnter New Price: ";
                                double TempPrice;
                                cin >> TempPrice;
                                if (cin.fail()) {
                                    cin.clear();
                                    cin.ignore();
                                    cout << "Please Enter A Valid Input this time: "
                                        << endl;
                                    Sleep(1000);
                                    goto RetryTempPrice;
                                }
                                Books[BookIndex].price = TempPrice;
                                cout << "Price Updated:" << endl;
                                cout << "Do You Want To update this Book Further: y/n";
                                cin >> LoopChoice;
                                break;
                            }
                            case 2: {
                            RetryTempQuantity:
                                system("cls");
                                cout << endl << endl << endl;
                                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2   Quantity Updating   "
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
```

```
                    cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                    cout << endl;
                    cout << "\t\tOld Quantity Of the Book Is :"
                        << Books[BookIndex].quantity << endl;
                    cout << "\t\tEnter New Quantity: ";
                    int TempQuantity;
                    cin >> TempQuantity;
                    if (cin.fail()) {
                        cin.clear();
                        cin.ignore();
                        cout << "Please Enter A Valid Input this time: "
                            << endl;
                        Sleep(1000);
                        goto RetryTempQuantity;
                    }
                    Books[BookIndex].quantity = TempQuantity;
                    cout << "Quantity Updated:" << endl;
                    cout << "Do You Want To update this Book Further: y/n";
                    cin >> LoopChoice;
                    break;
                }
                }
            }
            else {
                cout << "Please Input From given values: " <<
                    endl;
                Sleep(500);
                goto RetryDecision;
            }
        } while (LoopChoice == 'y');
    }
    else
        cout << "Book Not Found";
    File.open("TestBook.J");
    for (int i = 0; i < Index; i++) {
        File << " " << Books[i].name << endl
            << Books[i].authur << endl
            << Books[i].price << endl
            << Books[i].quantity << endl
            << endl;
    }
    File.close();
    system("pause");
}
else if (choice == 5) {
RetrySearch:
    system("cls");
    cout << endl << endl << endl;
    cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
        "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
        "\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
    cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
```

```
                    "\xb2      Book  Search     "
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xdb\xdb\xdb\n";
                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                cout << endl;
                cout << "\t\t1. Search Book By Name: " << endl;
                cout << "\t\t2. Search Book By Authur: " << endl;
                int SearchChoice;
                cin >> SearchChoice;
                if (cin.fail()) {
                    cin.clear();
                    cin.ignore();
                    cout << "Please Enter A Valid Input this time: "
                        << endl;
                    Sleep(1000);
                    goto RetrySearch;
                }
                if (SearchChoice == 1 || SearchChoice == 2) {
                    switch (SearchChoice) {
                    case 1: {
                        int BookIndex = SearchBook1();
                        if (BookFound != 0) {
                            cout << "Book Found: " << endl;
                            cout << "**********************************************************"
                                "*****************"
                                << endl;
                            cout << "Book Name: " << Books[BookIndex].name <<
                                "||"
                                << " Authur Name: " << Books[BookIndex].authur <<
                                "||"
                                << " Book Price: " << Books[BookIndex].price <<
                                "||"
                                << " Quantity: " << Books[BookIndex].quantity <<
                                endl;
                            cout << "**********************************************************"
                                "*****************"
                                << endl;
                            if (AuthurBookCount(Books[BookIndex].authur) ==
                                1) {
                                cout << "No Other Books Found By This Authur" <<
                                    endl;
                                cout << "******************************************************"
                                    "*********************"
                                    << endl;
                            }
                            else {
                                cout << AuthurBookCount(Books[BookIndex].authur)
                                    << " Books Found By this Authur: Press y to Show All His "
                                    "Books: "
                                    << endl;
                                cout << "******************************************************"
                                    "*********************"
                                    << endl;
                                char AuthurShow;
                                cin >> AuthurShow;
                                if (AuthurShow == 'y' || AuthurShow == 'Y') {
                                    AuthurShowBooks(Books[BookIndex].authur);
                                }
                            }
                        }
                        else
                            cout << "Book Not Found: ";
                        system("pause");
                        break;
                    }
                    case 2: {
                        SearchAuthur();
                        break;
                    }
                    }
                }
                else {
                    cout << "Please Enter From Given Numbers: " <<
                        endl;
                    Sleep(500);
                    goto RetrySearch;
                }
            }


            else if (choice == 6) {
                SaleBook();
            }
            else if (choice == 7) {
                ViewSoldBook();
            }
            else if (choice == 8) {
                return 0;
            }
            else if (choice == 4) {
                system("cls");
                cout << endl << endl << endl;
                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2    Book   Display     "
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xdb\xdb\xdb\n";
                cout << "\t\xdb\xdb\xdb\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
                    "\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
                cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
                    "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
                cout << endl;
                int BooksCount = 0;
                for (int i = 0; i < length - 1; i++) {
                    cout << "**********************************************************************"
                        "*******"
                        << endl;
                    cout << " Book Name:   " << Books[i].name << endl
                        << " Authur Name: " << Books[i].authur << endl
                        << " Book Price:  " << Books[i].price << endl
                        << " Quantity:    " << Books[i].quantity << endl;
                    BooksCount++;
                }
                cout << "**************************************************************************"
                    "***"
                    << endl;
                cout << BooksCount << " Books Found" << endl;
                cout << "**************************************************************************"
                    "***"
                    << endl;
                system("pause");
            }
        }
        else {
            cout << "Please Enter From Given Number: " <<
                endl;
            Sleep(1000);
        }
    }

    system("pause");
    return 0;
}
int main() {
    ReadData();
    ReadSoldData();
    system("color b2");
    system("title BOOK SALE MANAGMENT ");
    cout << endl << endl << endl;
    cout << "\n\n\t\t "
        "\xDB\xDB\xDB\xDB\xDB\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2"
        "\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xDB\xDB\xDB\xDB\xDB";
    cout << "\n\n\t\t \xDB\xDB\xDB\xDB\xDB\xB2 Book Sales Management\xB2\xDB\xDB\xDB\xDB\xDB";
    cout << "\n\n\t\t "
        "\xDB\xDB\xDB\xDB\xDB\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2"
        "\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xDB\xDB\xDB\xDB\xDB";
    cout << endl;
    cout << "        \t ";
    for (int i = 0; i < 11; i++) {
        Sleep(200);
        cout << "_";
        cout << "_";
        cout << "_";
        Sleep(200);
    }
    MainFunction();
}
void SaleBook() {
    int TempIndex = SearchBook();
retryQuant:
    system("cls");
    cout << "***************************************************************************"
        << endl;
    cout << "Book Name: " << Books[TempIndex].name <<
        " || "
        << " Authur Name: " << Books[TempIndex].authur <<
        " || "
        << " Book Price: " << Books[TempIndex].price <<
        " || "
        << " Quantity: " << Books[TempIndex].quantity <<
        endl;
    cout << "***************************************************************************"
        << endl;
    cout << "\t\tHow Many Books The custmer want to Buy: ";
    int TempQuant;
    cin >> TempQuant;
    if (cin.fail()) {
        cin.ignore();
        cin.clear();
        cout << "Enter a Valid Quantity";
        Sleep(500);
        goto retryQuant;
    }
  " || "Books[TempIndex].quantity
        Books[TempIndex].quantity =
            Books[TempIndex].quantity - TempQuant;
        SoldBooks[SaleIndex].name = Books[TempIndex].name;
        SoldBooks[SaleIndex].authur =
            Books[TempIndex].authur;
        SoldBooks[SaleIndex].price =
            Books[TempIndex].price;
        SoldBooks[SaleIndex].quantity = TempQuant;
        SaveSoldData();
        SaleIndex++;
        SaleLength++;
        cout << "\n\t\tThe Total Bill is: " <<
            (Books[TempIndex].price) * TempQuant;
        system("Pause");
        File.open("TestBook.J");
        for (int i = 0; i < Index; i++) {
            File << " " << Books[i].name << endl
                << Books[i].authur << endl
                << Books[i].price << endl
                << Books[i].quantity << endl
                << endl;
        }
        File.close();


    }
    else {
        cout << "Not enough Books Found: ";
    }
}
void SaveSoldData() {
    File.open("Salerecord.J", ios::app);
    for (int i = 0;
        i < SoldBooks[SaleIndex].name.length(); i++) {
        if (SoldBooks[SaleIndex].name[i] == 32) {
            SoldBooks[SaleIndex].name[i] = 95;
        }
    }
    for (int i = 0;
        i < SoldBooks[SaleIndex].authur.length(); i++) {
        if (SoldBooks[SaleIndex].authur[i] == 32) {
            SoldBooks[SaleIndex].authur[i] = 95;
        }
    }
    for (int i = length - 1; i < length; i++) {
        File << " " << SoldBooks[SaleIndex].name << endl
            << SoldBooks[SaleIndex].authur << endl
            << SoldBooks[SaleIndex].price << endl
            << SoldBooks[SaleIndex].quantity << endl
            << endl;
    }
    File.close();
}
void ReadSoldData() {
    Infile.open("Salerecord.J");
    for (int i = 0; i < SaleLength; i++) {
        while (!Infile.eof()) {

            Infile >> SoldBooks[SaleIndex].name;
            Infile >> SoldBooks[SaleIndex].authur;
            Infile >> SoldBooks[SaleIndex].price;
            Infile >> SoldBooks[SaleIndex].quantity;
            if (!Infile.eof()) {

                SaleIndex = SaleIndex + 1;
                SaleLength++;
            }
        }
        Infile.close();
    }
}
void ViewSoldBook() {
    int max = 0, Indexmax;
    for (int i = 0; i < SaleIndex; i++) {
        cout << "**********************************************************************"
            "*******"
            << endl;
        cout << " Book Name:   " << SoldBooks[i].name <<
            endl
            << " Authur Name: " << SoldBooks[i].authur << endl
            << " Book Price:  " << SoldBooks[i].price << endl
            << " Quantity:    " << SoldBooks[i].quantity <<
            endl;

        if (SoldBooks[i].quantity > max) {
            max = SoldBooks[i].quantity;
            Indexmax = i;
        }
    }
    cout << "**********************************************************************"
        "*******"
        << endl;
    cout << "\t\tThe Book With Maximum Sale" << endl;
    cout << "**********************************************************************"
        "*******"
        << endl;
    cout << "**********************************************************************"
        "*******"
        << endl;
    cout << " Book Name: " << SoldBooks[Indexmax].name
        << "||Authur Name: " << SoldBooks[Indexmax].authur
        << "||Book Price: " << SoldBooks[Indexmax].price <<
        " ||Quantity: " << SoldBooks[Indexmax].quantity <<
        endl;
    cout << "**********************************************************************"
        "*******"
        << endl;
    cout << "**********************************************************************"
        "*******"
        << endl;
    system("pause");
}
```