

```

1: #include <iostream>
2: #include <fstream>
3: #include <conio.h>
4: #include <string>
5: #include <windows.h>
6: #include <iomanip>
7: using namespace std;
8: struct Detail {
9:     string name;
10:    string authur;
11:    double price;
12:    int quantity;
13: };
14: int length = 1, Index = 0;
15: Detail Books[1000];
16: Detail SoldBooks[1000];
17: int SaleIndex = 0, SaleLength = 1;
18: int BookFound1 = 0;
19: int BookFound = 0;
20: int choice = 0;
21: int Design = 0;
22: ofstream File;
23: ifstream Infile;
24: void SaleBook();
25: void ViewSoldBook();
26: void SaveSoldData();
27: void ReadSoldData();
28: void MainFunction();
29: int main();
30: void SaveData() {
31:     File.open("TestBook.J", ios::app);
32:     for (int i = 0; i < Books[Index].name.length();
33:         i++) {
34:         if (Books[Index].name[i] == 32) {
35:             Books[Index].name[i] = 95;
36:         }
37:     }
38:     for (int i = 0; i < Books[Index].authur.length();
39:         i++) {
40:         if (Books[Index].authur[i] == 32) {
41:             Books[Index].authur[i] = 95;
42:         }
43:     }
44:     for (int i = length - 1; i < length; i++) {
45:         File << " " << Books[Index].name << endl
46:             << Books[Index].authur << endl
47:             << Books[Index].price << endl
48:             << Books[Index].quantity << endl
49:             << endl;
50:     }
51:     File.close();
52: }
53: void ReadData() {
54:     Infile.open("TestBook.J");
55:     while (!Infile.eof()) {
56:
57:         Infile >> Books[Index].name;
58:         Infile >> Books[Index].authur;
59:         Infile >> Books[Index].price;
60:         Infile >> Books[Index].quantity;
61:         if (!Infile.eof()) {

```

[illegible]

```

123:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
124:         "\n";
125:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
126:         "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
127:         "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
128:         "\n";
129:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
130:         "Book Search"
131:         "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
132:         "\n";
133:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
134:         "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
135:         "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
136:         "\n";
137:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
138:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
139:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
140:         "\n";
141:     cout << endl;
142:     cout << "\t\tEnter Book name: " << endl;
143:     string BookSearch;
144:
145:     getline(cin, BookSearch);
146:     " || "(int i = 0; i < BookSearch.length(); i++
147:         if (BookSearch[i] == 32) {
148:             BookSearch[i] = 95;
149:         }
150:     }
151:     for (int i = 0; i < SaleLength; i++) {
152:         if (BookSearch == SoldBooks[i].name) {
153:             retryS:
154:                 cout << "\t\tBook Found: ";
155:                 cout << "How Much Customer Want To Buy: ";
156:                 int tempQ;
157:                 cin >> tempQ;
158:                 if (cin.fail()) {
159:                     cin.ignore();
160:                     cin.clear();
161:                     cout << "Enter A Valid Quantity: ";
162:                     Sleep(500);
163:                     goto retryS;
164:                 }
165:                 SoldBooks[i].quantity = SoldBooks[i].quantity + tempQ;
166:                 File.open("Salerecord.J");
167:                 for (int i = 0; i < SaleIndex; i++) {
168:                     File << " " << SoldBooks[i].name << endl
169:                         << SoldBooks[i].author << endl
170:                         << SoldBooks[i].price << endl
171:                         << SoldBooks[i].quantity << endl
172:                         << endl;
173:                 }
174:                 File.close();
175:                 int Pricee;
176:                 for (int k = 0; k < length; k++) {
177:                     if (SoldBooks[i].name == Books[k].name) {
178:                         Books[k].quantity = Books[k].quantity - tempQ;
179:                         cout << "The Total Bill Is: " << Books[k].price * tempQ << endl;
180:                         break;
181:                     }
182:                 }
183:

```

```

184:         File.open("TestBook.J");
185:         for (int i = 0; i < Index; i++) {
186:             File << " " << Books[i].name << endl
187:                 << Books[i].author << endl
188:                 << Books[i].price << endl
189:                 << Books[i].quantity << endl
190:                 << endl;
191:         }
192:         File.close();
193:         system("pause");
194:         system("cls");
195:         MainFunction();
196:     }
197: }
198: for (int i = 0; i < length; i++) {
199:
200:     if (BookSearch == Books[i].name) {
201:         BookFound = 1;
202:         BookFound1 = 1;
203:         int Index = i;
204:         return Index;
205:     }
206: }
207: }
208: int AuthurBookCount(string Authur) {
209:     int AuthurCount = 0;
210:     for (int i = 0; i < length; i++) {
211:         if (Authur == Books[i].author) {
212:             AuthurCount++;
213:         }
214:     }
215:     return AuthurCount;
216: }
217: void AuthurShowBooks(string Authur) {
218:     system("cls");
219:     cout << endl << endl << endl;
220:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
221:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
222:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
223:         "\n";
224:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
225:         "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
226:         "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
227:         "\n";
228:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
229:         "Book Search "
230:         "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
231:         "\n";
232:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
233:         "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2"
234:         "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb"
235:         "\n";
236:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
237:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
238:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
239:         "\n";
240:     cout << endl;
241:     cout << "\t\tBooks By The Same Authur" << endl;
242:     for (int i = 0; i < length; i++) {
243:         if (Authur == Books[i].author) {
244:             cout << "*****"

```

[illegible]

[illegible]

[illegible]

```

428:         "\xb2\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
429:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\n";
430:     cout << "\\t\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
431:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
432:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
433:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
434:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
435:     cout << endl;
436:     Design = 0;
437: }
438: RetryTemp_Quantity:
439: if (Design) {
440:     cout << endl << endl << endl;
441:     cout << "\\t\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
442:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
443:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
444:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
445:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
446:     cout << "\\t\\xdb\\xdb\\xdb\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\n";
447:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\n";
448:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\n";
449:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\n";
450:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xdb\\xdb\\xdb\\n";
451:     cout << "\\t\\xdb\\xdb\\xdb\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\n";
452:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2      Book   Addition    "
453:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\n";
454:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xdb\\xdb\\xdb\\n";
455:     cout << "\\t\\xdb\\xdb\\xdb\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\n";
456:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\n";
457:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\n";
458:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\n";
459:     "\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xb2\\xdb\\xdb\\xdb\\n";
460:     cout << "\\t\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
461:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
462:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
463:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\db\\n";
464:     "\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\xdb\\n";
465:     cout << endl;
466:     Design = 0;
467: }
468: int Temp_Quantity;
469: cout << "\\t\\tCurrent Quantity of Books is: " <<
470:     Books[i].quantity
471:     << endl;
472: cout << "\\t\\tHow Much You Want to Add: ";
473: cin >> Temp_Quantity;
474: if (cin.fail()) {
475:     cin.clear();
476:     cin.ignore();
477:     cout << "Please Enter A Valid Input this time: "
478:         << endl;
479:     Sleep(1000);
480:     system("cls");
481:     Design = 1;
482:     goto RetryTemp_Quantity;
483: }
484: cin.ignore();
485: Books[i].quantity = Books[i].quantity +
486:     Temp_Quantity;
487: cout << "The New Quantity is: " <<
488:     Books[i].quantity << endl;

```



```

489:  " || "
490:      }
491:      File.open("TestBook.J");
492:      for (int i = 0; i < Index; i++) {
493:          File << " " << Books[i].name << endl
494:              << Books[i].author << endl
495:              << Books[i].price << endl
496:              << Books[i].quantity << endl
497:              << endl;
498:      }
499:      File.close();
500:  }
501:  }
502:  if (AuthurTaker) {
503:      lengthIn = 1;
504:      cout << "\t\tAuthur Name:      ";
505:      getline(cin, Books[Index].author);
506:      for (int i = 0; i < Books[Index].author.length();
507:          i++) {
508:          if (Books[Index].author[i] == 32) {
509:              Books[Index].author[i] = 95;
510:          }
511:      }
512:      RetryPrice:
513:      if (Design) {
514:          cout << endl << endl << endl;
515:          cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
516:              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
517:              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
518:              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
519:              "\xdb\xdb\xdb\n";
520:          cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
521:              "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
522:              "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
523:              "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
524:              "\xdb\xdb\xdb\n";
525:          cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
526:              "\xb2\xdb2\xdb2\xdb2      Book Addition      "
527:              "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
528:              "\xb2\xdb2\xdb2\xdb\xdb\xdb\n";
529:          cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
530:              "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
531:              "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
532:              "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
533:              "\xdb\xdb\xdb\n";
534:          cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
535:              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
536:              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
537:              "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
538:              "\xdb\xdb\xdb\n";
539:          cout << endl;
540:          Design = 0;
541:      }
542:      cout << "\t\tPrice of the Book:      ";
543:      cin >> Books[Index].price;
544:      if (cin.fail()) {
545:          cin.clear();
546:          cin.ignore();
547:          cout << "Please Enter A Valid Input this time: "
548:              << endl;
549:          Sleep(1000);

```

[illegible]

[illegible]

```
672:         Design = 0;
673:     }
674:     cout << "\t\t1. Delete Entire Book From System: "
675:           << endl;
676:     cout << "\t\t2. Reduce The Quantity Of Book: " <<
677:           endl;
678:     int DeleteChoice;
679:     cin >> DeleteChoice;
680:     if (cin.fail()) {
681:         cin.clear();
682:         cin.ignore();
683:         cout << "Please Enter A Valid Input: ";
684:         Sleep(250);
685:         system("cls");
686:         Design = 1;
687:         goto RetryDeleteDecision;
688:     }
689:     if (DeleteChoice == 1) {
690:         system("cls");
691:         Design = 1;
692:         if (Design) {
693:             cout << endl << endl << endl;
694:             cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
695:                   << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
696:                   << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
697:                   << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
698:                   << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
699:             cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
700:                   << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
701:                   << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
702:                   << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
703:                   << "\xb2\xdb2\xdb2\xdb2\xdb\xdb\xdb\n";
704:             cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
705:                   << "\xb2\xdb2\xdb2\xdb2\xdb2      Delete Book      "
706:                   << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
707:                   << "\xb2\xdb2\xdb2\xdb2\xdb\xdb\xdb\n";
708:             cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
709:                   << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
710:                   << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
711:                   << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
712:                   << "\xb2\xdb2\xdb2\xdb2\xdb\xdb\xdb\n";
713:             cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
714:                   << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
715:                   << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
716:                   << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
717:                   << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
718:             cout << endl;
719:             Design = 0;
720:         }
721:         for (int k = i; k < length; k++) {
722:             if (k + 1 <= length - 1) {
723:                 string temp = Books[k].name;
724:                 Books[k].name = Books[k + 1].name;
725:                 Books[k + 1].name = temp;
726:             }
727:             if (k + 1 <= length - 1) {
728:                 string temp = Books[k].author;
729:                 Books[k].author = Books[k + 1].author;
730:                 Books[k + 1].author = temp;
731:             }
732:             if (k + 1 <= length - 1) {
```

```

733:         float temp = Books[k].price;
734:         Books[k].price = Books[k + 1].price;
735:         Books[k + 1].price = temp;
736:     }
737:     if (k + 1 <= length - 1) {
738:         int temp = Books[k].quantity;
739:         Books[k].quantity = Books[k + 1].quantity;
740:         Books[k + 1].quantity = temp;
741:     }
742: }
743: length = length - 1;
744: Index = Index - 1;
745: BookFoundChecker++;
746: cout << "Book Deleted Successfully: " << endl;
747:
748: Sleep(500);
749: }
750: else if (DeleteChoice == 2) {
751:     system("cls");
752:     Design = 1;
753:     if (Design) {
754:         cout << endl << endl << endl;
755:         cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
756:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
757:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
758:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
759:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
760:         cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
761:             << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
762:             << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
763:             << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
764:             << "\xb2\xdb2\xdb2\xdb2\xdb\xdb\xdb\n";
765:         cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
766:             << "\xb2\xdb2\xdb2\xdb2\xdb2 Delete Book "
767:             << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
768:             << "\xb2\xdb2\xdb2\xdb2\xdb\xdb\xdb\n";
769:         cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
770:             << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
771:             << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
772:             << "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
773:             << "\xb2\xdb2\xdb2\xdb2\xdb\xdb\xdb\n";
774:         cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
775:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
776:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
777:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
778:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
779:         cout << endl;
780:         Design = 0;
781:     }
782:     cout << "\t\tThe Current Quantity Of the Book Named: "
783:         << Books[i].name
784:         << " is: " << Books[i].quantity << endl;
785: RetryTempDel:
786:     if (Design) {
787:         cout << endl << endl << endl;
788:         cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
789:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
790:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
791:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
792:             << "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
793:             << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"

```

```

794:         "\xb2\xb2\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
795:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
796:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
797:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
798:     cout << "\t\t\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
799:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
800:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
801:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
802:     cout << "\t\t\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
803:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
804:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
805:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
806:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
807:     cout << "\t\t\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
808:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
809:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
810:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
811:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
812:     cout << endl;
813:     Design = 0;
814: }
815: cout << "\t\tHow Much You Want To Reduce: ";
816: int TempDelQuantity;
817: cin >> TempDelQuantity;
818: if (cin.fail()) {
819:     cin.clear();
820:     cin.ignore();
821:     cout << "Enter A valid Input: ";
822:     Sleep(250);
823:     system("cls");
824:     Design = 1;
825:     goto RetryTempDel;
826: }
827: Books[i].quantity = Books[i].quantity -
828:     TempDelQuantity;
829: cout << "Quantity Updated: " << endl;
830: cout << "The New Quantity is: " <<
831:     Books[i].quantity << endl;
832: Sleep(300);
833: }
834: else {
835:     cout << "Please Enter From Given Number: " <<
836:         endl;
837:     Sleep(250);
838:     system("cls");
839:     Design = 1;
840:     goto RetryDeleteDecision;
841: }
842: }
843: }
844: if (BookFoundChecker == 0) {
845:     cout << "Book Not Found: ";
846: }
847: File.open("TestBook.J");
848: for (int i = 0; i < Index; i++) {
849:     File << " " << Books[i].name << endl
850:         << Books[i].author << endl
851:         << Books[i].price << endl
852:         << Books[i].quantity << endl
853:         << endl;
854: }

```

[illegible]

```

916:         "\xb2\xb2\xb2\xb2\b2\b2\b2\b2\b2\b2\b2"
917:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
918:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
919:     cout << "\t\xdb\xdb\xdb\b2\b2\b2\b2\b2\b2\b2\b2\b2"
920:         "\xb2\b2\b2\b2\b2\b2\b2      Price Updating      "
921:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
922:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
923:     cout << "\t\xdb\xdb\xdb\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
924:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
925:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
926:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
927:         "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
928:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
929:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
930:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
931:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
932:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
933:     cout << endl;
934:     cout << "\t\tThe Old Price Is : " <<
935:         Books[BookIndex].price
936:         << endl;
937:     cout << "\t\tEnter New Price: ";
938:     double TempPrice;
939:     cin >> TempPrice;
940:     if (cin.fail()) {
941:         cin.clear();
942:         cin.ignore();
943:         cout << "Please Enter A Valid Input this time: "
944:             << endl;
945:         Sleep(1000);
946:         goto RetryTempPrice;
947:     }
948:     Books[BookIndex].price = TempPrice;
949:     cout << "Price Updated:" << endl;
950:     cout << "Do You Want To update this Book Further: y/n";
951:     cin >> LoopChoice;
952:     break;
953: }
954: case 2: {
955:     RetryTempQuantity:
956:         system("cls");
957:         cout << endl << endl << endl;
958:         cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
959:             "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
960:             "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
961:             "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
962:             "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
963:         cout << "\t\xdb\xdb\xdb\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
964:             "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
965:             "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
966:             "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
967:             "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
968:         cout << "\t\xdb\xdb\xdb\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
969:             "\xb2\b2\b2\b2\b2\b2\b2      Quantity Updating      "
970:             "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
971:             "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
972:         cout << "\t\xdb\xdb\xdb\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
973:             "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
974:             "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
975:             "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"
976:             "\xb2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2\b2"

```



```

977:         cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
978:             "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
979:             "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
980:             "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
981:             "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
982:         cout << endl;
983:         cout << "\t\tOld Quantity Of the Book Is :";
984:         << Books[BookIndex].quantity << endl;
985:         cout << "\t\tEnter New Quantity: ";
986:         int TempQuantity;
987:         cin >> TempQuantity;
988:         if (cin.fail()) {
989:             cin.clear();
990:             cin.ignore();
991:             cout << "Please Enter A Valid Input this time: "
992:                 << endl;
993:             Sleep(1000);
994:             goto RetryTempQuantity;
995:         }
996:         Books[BookIndex].quantity = TempQuantity;
997:         cout << "Quantity Updated:" << endl;
998:         cout << "Do You Want To update this Book Further: y/n";
999:         cin >> LoopChoice;
1000:         break;
1001:     }
1002: }
1003: }
1004: else {
1005:     cout << "Please Input From given values: " <<
1006:         endl;
1007:     Sleep(500);
1008:     goto RetryDecision;
1009: }
1010: } while (LoopChoice == 'y');
1011: }
1012: else
1013:     cout << "Book Not Found";
1014: File.open("TestBook.J");
1015: for (int i = 0; i < Index; i++) {
1016:     File << " " << Books[i].name << endl
1017:         << Books[i].authur << endl
1018:         << Books[i].price << endl
1019:         << Books[i].quantity << endl
1020:         << endl;
1021: }
1022: File.close();
1023: system("pause");
1024: }
1025: else if (choice == 5) {
1026:     RetrySearch:
1027:     system("cls");
1028:     cout << endl << endl << endl;
1029:     cout << "\t\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
1030:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
1031:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb"
1032:         "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
1033:     cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
1034:         "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
1035:         "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"
1036:         "\xb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\n";
1037:     cout << "\t\xdb\xdb\xdb\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2\xdb2"

```

[illegible]

[illegible]

```

1160:         cout << "*****"
1161:         "*****"
1162:         << endl;
1163:         cout << " Book Name: " << Books[i].name << endl
1164:         << " Author Name: " << Books[i].author << endl
1165:         << " Book Price: " << Books[i].price << endl
1166:         << " Quantity: " << Books[i].quantity << endl;
1167:         BooksCount++;
1168:     }
1169:     cout << "*****"
1170:     "*****"
1171:     << endl;
1172:     cout << BooksCount << " Books Found" << endl;
1173:     cout << "*****"
1174:     "*****"
1175:     << endl;
1176:     system("pause");
1177: }
1178: }
1179: else {
1180:     cout << "Please Enter From Given Number: " <<
1181:     endl;
1182:     Sleep(1000);
1183: }
1184: }
1185:
1186: system("pause");
1187: return 0;
1188: }
1189: int main() {
1190:     ReadData();
1191:     ReadSoldData();
1192:     system("color b2");
1193:     system("title BOOK SALE MANAGMENT ");
1194:     cout << endl << endl << endl;
1195:     cout << "\n\n\t\t "
1196:     "\xDB\xDB\xDB\xDB\xDB\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2"
1197:     "\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2";
1198:     cout << "\n\n\t\t \xDB\xDB\xDB\xDB\xDB\xB2 Book Sales Management\xB2\xDB\xDB\xDB\xDB\xDB";
1199:     cout << "\n\n\t\t "
1200:     "\xDB\xDB\xDB\xDB\xDB\xDB\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2"
1201:     "\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2";
1202:     cout << endl;
1203:     cout << " \t ";
1204:     for (int i = 0; i < 11; i++) {
1205:         Sleep(200);
1206:         cout << "_";
1207:         cout << "_";
1208:         cout << "_";
1209:         Sleep(200);
1210:     }
1211:     MainFunction();
1212: }
1213: void SaleBook() {
1214:     int TempIndex = SearchBook();
1215:     retryQuant:
1216:     system("cls");
1217:     cout << "*****"
1218:     << endl;
1219:     cout << "Book Name: " << Books[TempIndex].name <<
1220:     " || "

```

```

1221:         << " Author Name: " << Books[TempIndex].author <<
1222:         " || "
1223:         << " Book Price: " << Books[TempIndex].price <<
1224:         " || "
1225:         << " Quantity: " << Books[TempIndex].quantity <<
1226:         endl;
1227:     cout << "*****"
1228:         << endl;
1229:     cout << "\t\tHow Many Books The customer want to Buy: ";
1230:     int TempQuant;
1231:     cin >> TempQuant;
1232:     if (cin.fail()) {
1233:         cin.ignore();
1234:         cin.clear();
1235:         cout << "Enter a Valid Quantity";
1236:         Sleep(500);
1237:         goto retryQuant;
1238:     }
1239:     " || "Books[TempIndex].quantity
1240:     Books[TempIndex].quantity =
1241:         Books[TempIndex].quantity - TempQuant;
1242:     SoldBooks[SaleIndex].name = Books[TempIndex].name;
1243:     SoldBooks[SaleIndex].author =
1244:         Books[TempIndex].author;
1245:     SoldBooks[SaleIndex].price =
1246:         Books[TempIndex].price;
1247:     SoldBooks[SaleIndex].quantity = TempQuant;
1248:     SaveSoldData();
1249:     SaleIndex++;
1250:     SaleLength++;
1251:     cout << "\n\t\tThe Total Bill is: " <<
1252:         (Books[TempIndex].price) * TempQuant;
1253:     system("Pause");
1254:     File.open("TestBook.J");
1255:     for (int i = 0; i < Index; i++) {
1256:         File << " " << Books[i].name << endl
1257:             << Books[i].author << endl
1258:             << Books[i].price << endl
1259:             << Books[i].quantity << endl
1260:             << endl;
1261:     }
1262:     File.close();
1263:
1264:
1265:     }
1266:     else {
1267:         cout << "Not enough Books Found: ";
1268:     }
1269: }
1270: void SaveSoldData() {
1271:     File.open("Salerecord.J", ios::app);
1272:     for (int i = 0;
1273:         i < SoldBooks[SaleIndex].name.length(); i++) {
1274:         if (SoldBooks[SaleIndex].name[i] == 32) {
1275:             SoldBooks[SaleIndex].name[i] = 95;
1276:         }
1277:     }
1278:     for (int i = 0;
1279:         i < SoldBooks[SaleIndex].author.length(); i++) {
1280:         if (SoldBooks[SaleIndex].author[i] == 32) {
1281:             SoldBooks[SaleIndex].author[i] = 95;

```

```

1282:     }
1283: }
1284: for (int i = length - 1; i < length; i++) {
1285:     File << " " << SoldBooks[SaleIndex].name << endl
1286:         << SoldBooks[SaleIndex].author << endl
1287:         << SoldBooks[SaleIndex].price << endl
1288:         << SoldBooks[SaleIndex].quantity << endl
1289:         << endl;
1290: }
1291: File.close();
1292: }
1293: void ReadSoldData() {
1294:     Infile.open("Salerecord.J");
1295:     for (int i = 0; i < SaleLength; i++) {
1296:         while (!Infile.eof()) {
1297:
1298:             Infile >> SoldBooks[SaleIndex].name;
1299:             Infile >> SoldBooks[SaleIndex].author;
1300:             Infile >> SoldBooks[SaleIndex].price;
1301:             Infile >> SoldBooks[SaleIndex].quantity;
1302:             if (!Infile.eof()) {
1303:
1304:                 SaleIndex = SaleIndex + 1;
1305:                 SaleLength++;
1306:             }
1307:         }
1308:         Infile.close();
1309:     }
1310: }
1311: void ViewSoldBook() {
1312:     int max = 0, Indexmax;
1313:     for (int i = 0; i < SaleIndex; i++) {
1314:         cout << "*****"
1315:             << endl;
1316:         cout << " Book Name: " << SoldBooks[i].name <<
1317:             endl
1318:             << " Author Name: " << SoldBooks[i].author << endl
1319:             << " Book Price: " << SoldBooks[i].price << endl
1320:             << " Quantity: " << SoldBooks[i].quantity <<
1321:             endl;
1322:         if (SoldBooks[i].quantity > max) {
1323:             max = SoldBooks[i].quantity;
1324:             Indexmax = i;
1325:         }
1326:     }
1327: }
1328: cout << "*****"
1329:     << endl;
1330: cout << "\t\tThe Book With Maximum Sale" << endl;
1331: cout << "*****"
1332:     << endl;
1333: cout << "*****"
1334:     << endl;
1335: cout << "*****"
1336:     << endl;
1337: cout << " Book Name: " << SoldBooks[Indexmax].name
1338:     << " || Author Name: " << SoldBooks[Indexmax].author
1339:     << " || Book Price: " << SoldBooks[Indexmax].price <<
1340:     << " || Quantity: " << SoldBooks[Indexmax].quantity <<

```

```
1343:         endl;
1344:     cout << "*****"
1345:         "*****"
1346:         << endl;
1347:     cout << "*****"
1348:         "*****"
1349:         << endl;
1350:     system("pause");
1351: }
```