

# Master Code

## Algorithm

- **Initialization (Setup):**
  - Define the **Book** data structure (ID, Title, Author, Price, Quantity).
  - Declare global arrays: **books[]** (stores inventory) and **cart[]** (stores user selections).
  - Initialize global counters: **bookCount = 0** and **cartCount = 0**.
- **Startup (Load Data):**
  - Check for the existence of **library.dat** (the storage file).
  - If the file exists: Read the data into the **books[]** array and update **bookCount**.
  - If the file does not exist: Start with an empty library.
- **Main Program Loop (Controller):**
  - Enter an infinite loop to keep the system running.
  - Display the **Main Menu** (Visitor, Admin, Exit).
  - Accept the user's choice.
- **Routing Logic (Dispatcher):**
  - **If Visitor is selected:**
    - Call the **Visitor Panel Subroutine**.
    - Allow browsing, searching, cart management, and purchasing.
    - Return to the Main Menu when finished.
  - **If Admin is selected:**
    - Call the **Admin Panel Subroutine**.
    - Prompt for Username and Password.
    - If valid: Allow adding or deleting books.
    - Return to the Main Menu when finished.
  - **If Exit is selected:**
    - Call the **Save to File** function to write current **books[]** data to **library.dat**.
    - Break the loop.
- **Termination:**
  - End the program.

## Pseudo Code

```
// --- GLOBAL DEFINITIONS ---  
CONSTANT MAX_BOOKS = 100  
CONSTANT MAX_CART = 50  
CONSTANT FILENAME = "library.dat"
```

```
STRUCTURE Book  
    INTEGER id  
    STRING title  
    STRING author  
    FLOAT price  
    INTEGER quantity  
END STRUCTURE
```

```
// Global Variables  
Book books[MAX_BOOKS]
```

```
INTEGER cart[MAX_CART]
```

```
INTEGER bookCount = 0
```

```
INTEGER cartCount = 0
```

```
// --- MASTER PROGRAM START ---
```

```
START
```

```
    // 1. Initialize System
```

```
    CALL LoadDataFromFile()
```

```
    // 2. Main Control Loop
```

```
    WHILE (True) DO
```

```
        PRINT "==== Library Management System ====="
```

```
        PRINT "1. Visitor"
```

```
        PRINT "2. Admin"
```

```
        PRINT "0. Exit"
```

```
        READ choice
```

```
        SWITCH (choice)
```

```
            CASE 1: // VISITOR PATH
```

```
                CALL VisitorPanel()
```

```
                BREAK
```

```
            CASE 2: // ADMIN PATH
```

```
                CALL AdminPanel()
```

```
                BREAK
```

```
            CASE 0: // EXIT PATH
```

```
                CALL SaveDataToFile()
```

```
                EXIT WHILE LOOP
```

```
            DEFAULT:
```

```
                PRINT "Invalid Selection. Please try again."
```

```
        END SWITCH
```

```
    END WHILE
```

```
    // 3. Finalize
```

```
    PRINT "System Shutting Down..."
```

```
END
```

```
// --- SUBROUTINE DEFINITIONS ---
```

```
// --- DATA PERSISTENCE ---
```

```
PROCEDURE LoadDataFromFile()
```

```
    IF (FileExists(FILENAME)) THEN
```

```
    OPEN FILENAME for READING
    READ bookCount
    FOR i = 0 TO bookCount DO
        READ books[i]
    END FOR
    CLOSE FILE
END IF
END PROCEDURE
```

```
PROCEDURE SaveDataToFile()
    OPEN FILENAME for WRITING
    WRITE bookCount
    FOR i = 0 TO bookCount DO
        WRITE books[i]
    END FOR
    CLOSE FILE
END PROCEDURE
```

```
// --- VISITOR MODULE ---
```

```
PROCEDURE VisitorPanel()
    WHILE (True) DO
        PRINT "1. View | 2. Search | 3. Add Cart | 4. Rem Cart | 5. View Cart | 6. Buy | 0. Back"
        READ vChoice

        IF (vChoice == 1) THEN
            CALL ViewAllBooks()
        ELSE IF (vChoice == 2) THEN
            CALL SearchBooks()
        ELSE IF (vChoice == 3) THEN
            CALL AddToCart()
        ELSE IF (vChoice == 4) THEN
            CALL RemoveFromCart()
        ELSE IF (vChoice == 5) THEN
            CALL DisplayCartTotal()
        ELSE IF (vChoice == 6) THEN
            CALL ProcessPurchase() // Decrements stock and saves file
        ELSE IF (vChoice == 0) THEN
            RETURN // Back to Main Menu
        END IF
    END WHILE
END PROCEDURE
```

```
// --- ADMIN MODULE ---
```

```
PROCEDURE AdminPanel()
    PRINT "Enter Username: "
    READ user
```

```

PRINT "Enter Password: "
READ pass

IF (user != "admin" OR pass != "1234") THEN
    PRINT "Access Denied"
    RETURN
END IF

WHILE (True) DO
    PRINT "1. Add Book | 2. Delete Book | 0. Logout"
    READ aChoice

    IF (aChoice == 1) THEN
        CALL AddBook() // Inputs data and saves file
    ELSE IF (aChoice == 2) THEN
        CALL DeleteBook() // Removes data and saves file
    ELSE IF (aChoice == 0) THEN
        RETURN // Back to Main Menu
    END IF
END WHILE
END PROCEDURE

```

## Main Program Flow

### Algorithm:

**START** the program.

**Load** existing book data from the storage file into the **books** array. If no file exists, start with an empty list.

**Enter an Infinite Loop** to keep the program running until explicitly exited.

**Display** the Main Menu options:

1. Visitor
2. Admin
3. Exit

**Read** the user's input choice.

**Check** the choice:

- **If 1:** Transfer control to the **Visitor Panel**.
- **If 2:** Transfer control to the **Admin Panel**.
- **If 0:** Save the current state of books to the file, then **Exit** the loop.
- **Otherwise:** Display an "Invalid Input" error.

**Repeat** the loop from step 4.

**END** the program.

### Pseudo Code:

```

START
    INITIALIZE bookCount = 0, cartCount = 0
    CALL LoadFromFile()

```

```
WHILE (True) DO
    PRINT "==== Library Management System ====="
    PRINT "1. Visitor"
    PRINT "2. Admin"
    PRINT "0. Exit"
    READ choice

    IF (choice == 1) THEN
        CALL VisitorPanel()
    ELSE IF (choice == 2) THEN
        CALL AdminPanel()
    ELSE IF (choice == 0) THEN
        CALL SaveToFile()
        BREAK
    ELSE
        PRINT "Invalid selection!"
    END IF
END WHILE
END
```

## Visitor Panel Subroutine

### Algorithm

**START** Visitor Panel.

**Enter an Infinite Loop.**

**Display** the Visitor Menu (View, Search, Add Cart, Remove, View Cart, Confirm, Back).

**Read** user choice.

**Process** choice:

- **View Books:** Loop through the **books** array and print every book's details.
- **Search:** Read a keyword string. Loop through books and print details if the keyword matches Title or Author (case-insensitive).
- **Add to Cart:** Read Book ID. Search **books** array.
  - If found and Quantity > 0: Add ID to **cart** array.
  - Else: Print error.
- **Remove from Cart:** Read ID. Search **cart** array, remove the item, and shift array elements.
- **View Cart:** Loop through **cart** array, find corresponding book details, calculate total price, and print.
- **Confirm Purchase:** Loop through **cart** array.
  - Find corresponding book in **books** array.
  - Decrement the book **quantity**.
  - Save data to file.
  - Clear the **cart** array.

**Back:** Break loop and return to Main Menu.

**END** Visitor Panel.

### Pseudo Code:

PROCEDURE VisitorPanel()

  WHILE (True) DO

    PRINT "==== Visitor Panel ====="

    PRINT "1. View Books"

    PRINT "2. Search Book"

    PRINT "3. Add to Cart"

    PRINT "4. Remove from Cart"

    PRINT "5. View Cart"

    PRINT "6. Confirm Purchase"

    PRINT "0. Back"

    READ choice

  IF (choice == 1) THEN

    FOR i = 0 TO bookCount DO

      PRINT books[i].details

    END FOR

  ELSE IF (choice == 2) THEN

    READ keyword

    FOR i = 0 TO bookCount DO

      IF (keyword MATCHES books[i].title OR books[i].author) THEN

```

        PRINT books[i].details
    END IF
END FOR

ELSE IF (choice == 3) THEN
    READ id
    FOR i = 0 TO bookCount DO
        IF (books[i].id == id) THEN
            IF (books[i].quantity > 0) THEN
                cart[cartCount] = id
                INCREMENT cartCount
                PRINT "Added!"
            ELSE
                PRINT "Out of stock"
            END IF
        END IF
    END FOR

ELSE IF (choice == 4) THEN
    READ id
    FOR i = 0 TO cartCount DO
        IF (cart[i] == id) THEN
            REMOVE cart[i] and SHIFT remaining items left
            DECREMENT cartCount
            PRINT "Removed"
        END IF
    END FOR

ELSE IF (choice == 5) THEN
    total = 0
    FOR i = 0 TO cartCount DO
        Find book details for cart[i]
        PRINT book.title, book.price
        total = total + book.price
    END FOR
    PRINT total

ELSE IF (choice == 6) THEN
    FOR i = 0 TO cartCount DO
        Find book in books[] matching cart[i]
        DECREMENT book.quantity
    END FOR
    CALL SaveToFile()
    cartCount = 0
    PRINT "Purchase Successful"

ELSE IF (choice == 0) THEN
    BREAK

```

```
END IF
END WHILE
END PROCEDURE
```

## Admin Panel Subroutine

### Algorithm

**START** Admin Panel.

**Read** Username and Password.

**Verify** credentials (Hardcoded "admin" / "1234").

**If Invalid:** Print "Access Denied" and return to Main Menu.

**If Valid:** Enter an Infinite Loop.

**Display** Admin Menu (Add, Delete, Logout).

**Read** choice.

**Process** choice:

- **Add Book:** Read Title, Author, Price, Quantity. Generate new ID. Add new structure to **books** array. Save to file.
- **Delete Book:** Read ID. Search **books** array.
  - If found: Shift all subsequent books left to overwrite it. Decrement **bookCount**. Save to file.
- **Logout:** Break loop and return to Main Menu.

**END** Admin Panel.

### Pseudo Code

```
PROCEDURE AdminPanel()
```

```
    PRINT "User: "
```

```
    READ inputUser
```

```
    PRINT "Pass: "
```

```
    READ inputPass
```

```
    IF (inputUser != "admin" OR inputPass != "1234") THEN
```

```
        PRINT "Invalid Credentials"
```

```
        RETURN
```

```
    END IF
```

```
    WHILE (True) DO
```

```
        PRINT "-Admin-"
```

```
        PRINT "1. Add Book"
```

```
        PRINT "2. Delete Book"
```

```
        PRINT "0. Logout"
```

```
        READ choice
```

```
        IF (choice == 1) THEN
```

```
            IF (bookCount > 0) THEN
```

```
                newId = books[bookCount-1].id + 1
```

```
            ELSE
```

```
                newId = 1
```

```
            END IF
```

READ title, author, price, qty

books[bookCount].id = newId  
books[bookCount].title = title  
books[bookCount].author = author  
books[bookCount].price = price  
books[bookCount].quantity = qty

INCREMENT bookCount  
CALL SaveToFile()

ELSE IF (choice == 2) THEN  
 READ id  
 FOR i = 0 TO bookCount DO  
 IF (books[i].id == id) THEN  
 // Shift elements to left to delete  
 FOR j = i TO bookCount - 2 DO  
 books[j] = books[j+1]  
 END FOR  
 DECREMENT bookCount  
 CALL SaveToFile()  
 BREAK  
 END IF  
 END FOR

ELSE IF (choice == 0) THEN  
 BREAK  
END IF  
END WHILE  
END PROCEDURE