Library Management System - Complete C Code

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_BOOKS 100

// Book structure for the array
struct Book {
    int id;
    char title[100];
    char author[100];
    int isIssued;
};

// Linked list node for issued books
struct IssuedBook {
    int id;
    struct IssuedBook* next;
};

// Global array of books
struct Book library[MAX_BOOKS];
int bookCount = 0;

// Head of issued books linked list
struct IssuedBook* issuedHead = NULL;

// Add a book to the array
void addBook(int id, const char* title, const char* author) {
    if (bookCount >= MAX_BOOKS) {
        printf("Library is full. Cannot add more books.\n");
        return;
    }

    library[bookCount].id = id;
    strcpy(library[bookCount].title, title);
    strcpy(library[bookCount].author, author);
    library[bookCount].isIssued = 0;
    bookCount++;
    printf("Book added successfully.\n");
}

// Delete a book from the array
void deleteBook(int id) {
    int found = 0;
    for (int i = 0; i < bookCount; i++) {
        if (library[i].id == id) {
            if (library[i].isIssued) {
                printf("Cannot delete book with ID %d as it is currently issued.\n", id);
                return;
            }

            found = 1;
            for (int j = i; j < bookCount - 1; j++) {
                library[j] = library[j + 1];
            }
            bookCount--;
            printf("Book with ID %d deleted successfully.\n", id);
            break;
        }
    }
    if (!found) {
        printf("Book with ID %d not found.\n", id);
    }
}

// Search for a book by ID
void searchBook(int id) {
    for (int i = 0; i < bookCount; i++) {
        if (library[i].id == id) {
            printf("\nBook Found:\n");
            printf("ID: %d\n", library[i].id);
            printf("Title: %s\n", library[i].title);
            printf("Author: %s\n", library[i].author);
            printf("Status: %s\n", library[i].isIssued ? "Issued" : "Available");
            return;
        }
    }
    printf("Book with ID %d not found.\n", id);
}

// Display all books
void displayBooks() {
    if (bookCount == 0) {
        printf("No books in the library.\n");
        return;
    }

    printf("\nLibrary Book List:\n");
    printf("-------------------------------------\n");
    printf("ID   | Title           | Author      | Status\n");
    printf("-------------------------------------\n");
    for (int i = 0; i < bookCount; i++) {
        printf("%-4d | %-15s | %-10s | %-8s\n",
               library[i].id,
               library[i].title,
               library[i].author,
               library[i].isIssued ? "Issued" : "Available");
    }
    printf("-------------------------------------\n");
}
```

Library Management System - Complete C Code

```c
// Issue a book
void issueBook(int id) {
    for (int i = 0; i < bookCount; i++) {
        if (library[i].id == id && !library[i].isIssued) {
            struct IssuedBook* newIssued = (struct IssuedBook*)malloc(sizeof(struct IssuedBook));
            newIssued->id = id;
            newIssued->next = issuedHead;
            issuedHead = newIssued;
            library[i].isIssued = 1;
            printf("Book with ID %d issued successfully.\n", id);
            return;
        }
    }
    printf("Book not available or already issued.\n");
}

// Return/Unissue a book
void returnBook(int id) {
    struct IssuedBook* temp = issuedHead;
    struct IssuedBook* prev = NULL;

    while (temp != NULL) {
        if (temp->id == id) {
            if (prev == NULL) {
                issuedHead = temp->next;
            } else {
                prev->next = temp->next;
            }
            free(temp);

            for (int i = 0; i < bookCount; i++) {
                if (library[i].id == id) {
                    library[i].isIssued = 0;
                    break;
                }
            }

            printf("Book with ID %d returned successfully.\n", id);
            return;
        }
        prev = temp;
        temp = temp->next;
    }

    printf("Book with ID %d is not issued.\n", id);
}

// Display issued books
void displayIssuedBooks() {
    struct IssuedBook* temp = issuedHead;
    if (!temp) {
        printf("No books have been issued.\n");
        return;
    }

    printf("\nIssued Book List:\n");
    while (temp != NULL) {
        printf("ID: %d\n", temp->id);
        temp = temp->next;
    }
}

// Free allocated memory before exiting
void freeIssuedBooks() {
    struct IssuedBook* temp;
    while (issuedHead != NULL) {
        temp = issuedHead;
        issuedHead = issuedHead->next;
        free(temp);
    }
}

// Menu
void menu() {
    int choice, id;
    char title[100], author[100];

    while (1) {
        printf("\nLibrary Management System\n");
        printf("1. Add Book\n");
        printf("2. Delete Book\n");
        printf("3. Search Book\n");
        printf("4. Display All Books\n");
        printf("5. Issue Book\n");
        printf("6. Return Book\n");
        printf("7. Display Issued Books\n");
        printf("8. Exit\n");
        printf("Enter choice: ");

        if (scanf("%d", &choice) != 1) {
            printf("Invalid input! Please enter a number.\n");
            while (getchar() != '\n');
            continue;
        }

        switch (choice) {
            case 1:
                printf("Enter Book ID: ");
                scanf("%d", &id);
                getchar();
                printf("Enter Title: ");
                fgets(title, sizeof(title), stdin);
                title[strcspn(title, "\n")] = '\0';
                printf("Enter Author: ");
                fgets(author, sizeof(author), stdin);
                author[strcspn(author, "\n")] = '\0';
                addBook(id, title, author);
                break;
            case 2:
                printf("Enter Book ID to delete: ");
                scanf("%d", &id);
                deleteBook(id);
                break;
            case 3:
                printf("Enter Book ID to search: ");
                scanf("%d", &id);
                searchBook(id);
                break;
            case 4:
                displayBooks();
                break;
            case 5:
                printf("Enter Book ID to issue: ");
                scanf("%d", &id);
                issueBook(id);
                break;
            case 6:
                printf("Enter Book ID to return: ");
                scanf("%d", &id);
                returnBook(id);
                break;
            case 7:
                displayIssuedBooks();
                break;
            case 8:
                freeIssuedBooks();
                printf("Exiting...\n");
                exit(0);
            default:
                printf("Invalid choice. Try again.\n");
        }
    }
}

int main() {
    menu();
    return 0;
}
```