

C Program: Book Management System

This C program will allow users to manage a collection of book records. The program will include the following functionalities:

- **Add a book record**
- **Search for a book**
- **Delete a book record**
- **Display all book records**

The program will use an array of structures to store the book records, and menu options will be provided for the user to perform different operations on the records.

Book Structure:

Each book will have the following attributes:

- **Book ID** (unique identifier for each book)
- **Book Title**
- **Author Name**
- **Publisher**
- **Year of Publication**

C Program: Book Management System

```
#include <stdio.h>
```

```
#include <string.h>
```

```
// Define structure to represent a book
```

```
struct Book {
```

```
    int id;
```

```
    char title[100];
```

```
    char author[100];
```

```
    char publisher[100];
```

```
    int year;
```

```
};
```

```
// Function to add a new book
```

```
void addBook(struct Book library[], int *count) {
```

```
    struct Book newBook;
```

```
    printf("Enter book ID: ");
```

```
    scanf("%d", &newBook.id);
```

```
    getchar(); // to consume newline after reading ID
```

```
    printf("Enter book title: ");
```

```
    fgets(newBook.title, sizeof(newBook.title), stdin);
```

```
    newBook.title[strcspn(newBook.title, "\n")] = '\0'; // remove newline
```

```
    printf("Enter author name: ");
```

```
    fgets(newBook.author, sizeof(newBook.author), stdin);
```

```
    newBook.author[strcspn(newBook.author, "\n")] = '\0'; // remove newline
```

```
    printf("Enter publisher name: ");
```

```
    fgets(newBook.publisher, sizeof(newBook.publisher), stdin);
```

```
    newBook.publisher[strcspn(newBook.publisher, "\n")] = '\0'; // remove newline
```

```
    printf("Enter year of publication: ");
```

```
    scanf("%d", &newBook.year);
```

```
    // Add book to library
```

```
    library[*count] = newBook;
```

```
    (*count)++;
```

```
    printf("Book added successfully!\n");
```

```
}
```

```
// Function to search for a book by title
```

```
void searchBookByTitle(struct Book library[], int count) {
```

```
    char searchTitle[100];
```

```
    getchar(); // consume the leftover newline character
```

```
    printf("Enter the title of the book to search: ");
```

```

fgets(searchTitle, sizeof(searchTitle), stdin);
searchTitle[strcspn(searchTitle, "\n")] = '\0'; // remove newline

int found = 0;
for (int i = 0; i < count; i++) {
    if (strstr(library[i].title, searchTitle) != NULL) {
        printf("Book found: \n");
        printf("ID: %d\n", library[i].id);
        printf("Title: %s\n", library[i].title);
        printf("Author: %s\n", library[i].author);
        printf("Publisher: %s\n", library[i].publisher);
        printf("Year: %d\n", library[i].year);
        found = 1;
    }
}
if (!found) {
    printf("No book found with the title: %s\n", searchTitle);
}
}

// Function to delete a book by ID
void deleteBookById(struct Book library[], int *count) {
    int id;
    printf("Enter book ID to delete: ");
    scanf("%d", &id);

    int found = 0;
    for (int i = 0; i < *count; i++) {
        if (library[i].id == id) {
            // Shift all books after the deleted one
            for (int j = i; j < *count - 1; j++) {
                library[j] = library[j + 1];
            }
            (*count)--; // Reduce book count
            printf("Book with ID %d deleted successfully.\n", id);
            found = 1;
        }
    }
}

```

```

        break;
    }
}
if (!found) {
    printf("No book found with ID %d.\n", id);
}
}

```

```

// Function to display all books
void displayAllBooks(struct Book library[], int count) {
    if (count == 0) {
        printf("No books available.\n");
        return;
    }

```

```

    printf("\nAll Books in Library:\n");
    for (int i = 0; i < count; i++) {
        printf("\nID: %d\n", library[i].id);
        printf("Title: %s\n", library[i].title);
        printf("Author: %s\n", library[i].author);
        printf("Publisher: %s\n", library[i].publisher);
        printf("Year: %d\n", library[i].year);
    }
}

```

```

// Main function to provide the menu and manage operations
int main() {
    struct Book library[100]; // Array to store books (maximum 100 books)
    int bookCount = 0;        // To track the number of books

    int choice;
    do {
        // Display menu
        printf("\n--- Book Management System ---\n");
        printf("1. Add Book\n");
        printf("2. Search Book by Title\n");

```

```

printf("3. Delete Book by ID\n");
printf("4. Display All Books\n");
printf("5. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);

// Perform action based on user's choice
switch (choice) {
    case 1:
        addBook(library, &bookCount);
        break;
    case 2:
        searchBookByTitle(library, bookCount);
        break;
    case 3:
        deleteBookById(library, &bookCount);
        break;
    case 4:
        displayAllBooks(library, bookCount);
        break;
    case 5:
        printf("Exiting the program...\n");
        break;
    default:
        printf("Invalid choice, please try again.\n");
}
} while (choice != 5);

return 0;
}

```

Explanation of the Program:

1. Book Structure:

- We define a **struct Book** that holds information about each book, including its ID, title, author, publisher, and year of publication.

2. **Add Book** (`addBook`):
 - This function allows the user to enter details for a new book (ID, title, author, publisher, and year) and stores the book in the `library` array.
 - The `count` of books is updated after adding a new book.
 3. **Search Book by Title** (`searchBookByTitle`):
 - This function prompts the user to enter a book title to search for.
 - It searches for books whose titles contain the search term (using `strstr()` function).
 - If a match is found, the details of the book are displayed. If no match is found, it displays a "no book found" message.
 4. **Delete Book by ID** (`deleteBookById`):
 - This function prompts the user to enter the ID of the book to delete.
 - If a book with the entered ID is found, it removes the book by shifting all subsequent books down by one position in the array. The book count is decremented.
 - If no matching book is found, an error message is displayed.
 5. **Display All Books** (`displayAllBooks`):
 - This function displays the details of all books currently stored in the `library` array.
 - If there are no books in the library, it prints a message saying "No books available."
 6. **Menu and Loop**:
 - The main function provides a simple text-based menu where the user can choose which operation to perform (add, search, delete, display).
 - The program keeps running until the user selects the option to exit (option 5).
-

Sample Output:

Menu and Adding a Book:

--- Book Management System ---

1. Add Book
2. Search Book by Title
3. Delete Book by ID
4. Display All Books

5. Exit

Enter your choice: 1

Enter book ID: 101

Enter book title: C Programming Guide

Enter author name: John Doe

Enter publisher name: XYZ Publishers

Enter year of publication: 2020

Book added successfully!

Searching for a Book:

--- Book Management System ---

1. Add Book

2. Search Book by Title

3. Delete Book by ID

4. Display All Books

5. Exit

Enter your choice: 2

Enter the title of the book to search: C Programming

Book found:

ID: 101

Title: C Programming Guide

Author: John Doe

Publisher: XYZ Publishers

Year: 2020

Deleting a Book:

--- Book Management System ---

1. Add Book

2. Search Book by Title

3. Delete Book by ID

4. Display All Books

5. Exit

Enter your choice: 3

Enter book ID to delete: 101

Book with ID 101 deleted successfully.

Displaying All Books:

--- Book Management System ---

1. Add Book
2. Search Book by Title
3. Delete Book by ID
4. Display All Books
5. Exit

Enter your choice: 4

All Books in Library:

ID: 102

Title: Advanced C Programming

Author: Jane Smith

Publisher: ABC Publishers

Year: 2022

Exiting the Program:

--- Book Management System ---

1. Add Book
2. Search Book by Title
3. Delete Book by ID
4. Display All Books
5. Exit

Summary of Program Logic:

1. **Menu-driven Interface:** The user is presented with a menu to choose an operation (Add, Search, Delete, Display, Exit).
2. **Add Book:** New books are added to the **library** array.
3. **Search Book:** The program searches for books by title, using substring matching.

4. **Delete Book:** Books can be deleted by ID, and the array is updated to remove the deleted book.
5. **Display Books:** All books in the library are displayed.

Extending the Program:

- You could add more features, such as sorting books by title or author, updating book details, or saving/loading the book data from a file.
- For a large number of books, dynamic memory allocation (using pointers) could be used instead of a fixed-size array.