DOCMENTATION

Abstract

A Library Management System is a computerized system designed to manage and organize library resources efficiently. It helps in maintaining records of books, members, issue and return details, and availability status. By automating routine library operations, the system reduces manual work, minimizes errors, and improves the overall efficiency of library management. This system provides quick access to information and ensures proper utilization of library resources.

Introduction

Libraries play an important role in educational institutions by providing access to knowledge and learning materials. Traditional library management methods involve manual record keeping, which is time-consuming and prone to errors. The Library Management System overcomes these limitations by using computer technology to store, retrieve, and manage library data. It allows librarians to manage books, members, and transactions easily while providing faster and more accurate services to users.

Software Requirements

Operating System: Windows / Linux

Programming Language: C / Java / Python (as per implementation)

Database: MySQL / File handling system

Development Tools: CodeBlocks / Visual Studio / Eclipse

MS Office (for documentation and reports)

Hardware Requirements

Processor: Intel i3 or above

RAM: Minimum 4 GB

Hard Disk: 500 GB or higher

Keyboard and Mouse

Monitor

Printer (optional, for reports and receipts)

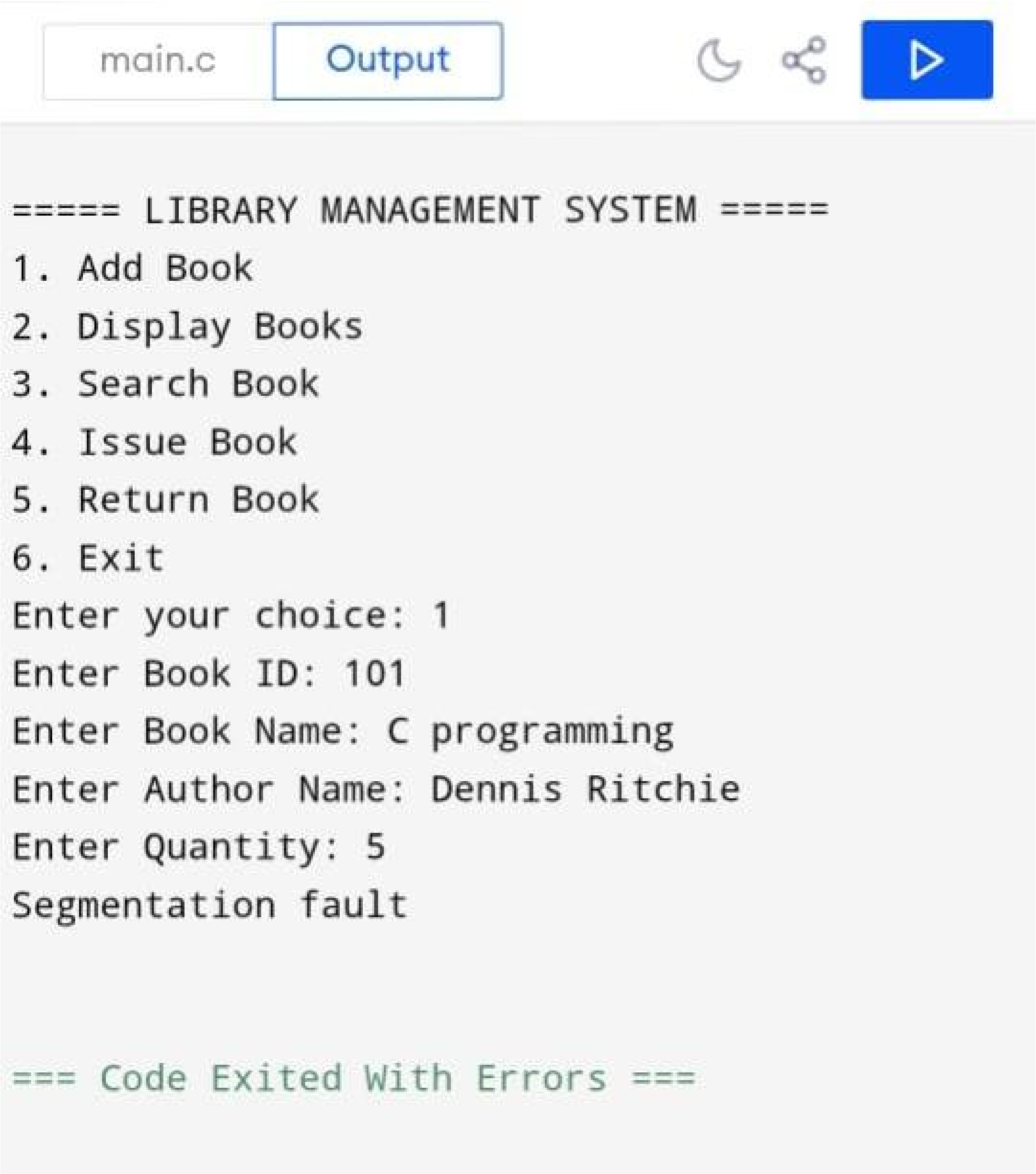
Code

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

struct book
{
   int id;
   char name[50];
   char author[50];
   int quantity;
};

void addBook();
void displayBooks();
void searchBook();
void issueBook();
void returnBook();

int main()
{
    int choice;

    while (1)
    {
        printf("\n===== LIBRARY MANAGEMENT SYSTEM =====\n");
        printf("1. Add Book\n");
        printf("2. Display Books\n");
        printf("3. Search Book\n");
        printf("4. Issue Book\n");
        printf("5. Return Book\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            addBook();
            break;
        case 2:
            displayBooks();
            break;
        case 3:
            searchBook();
            break;
        case 4:
            issueBook();
            break;
        case 5:
            returnBook();
            break;
        case 6:
            exit(0);
        default:
            printf("Invalid choice!\n");
        }
    }
    return 0;
}

/* Add Book */
void addBook()
{
    struct book b;
    FILE *fp = fopen("library.dat", "ab");

    printf("Enter Book ID: ");
    scanf("%d", &b.id);
    printf("Enter Book Name: ");
    scanf(" %[^\n]", b.name);
    printf("Enter Author Name: ");
    scanf(" %[^\n]", b.author);
    printf("Enter Quantity: ");
    scanf("%d", &b.quantity);

    fwrite(&b, sizeof(b), 1, fp);
    fclose(fp);

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

/* Display Books */
void displayBooks()
{
    struct book b;
    FILE *fp = fopen("library.dat", "rb");

    if (fp == NULL)
    {
        printf("No books available.\n");
        return;
    }

    printf("\nID\tName\t\tAuthor\t\tQuantity\n");
    while (fread(&b, sizeof(b), 1, fp))
    {
        printf("%d\t%s\t\t%s\t\t%d\n",
               b.id, b.name, b.author, b.quantity);
    }
    fclose(fp);
}

/* Search Book */
void searchBook()
{
    struct book b;
    int id, found = 0;
    FILE *fp = fopen("library.dat", "rb");

    printf("Enter Book ID to search: ");
    scanf("%d", &id);

    while (fread(&b, sizeof(b), 1, fp))
    {
        if (b.id == id)
        {
            printf("Book Found!\n");
            printf("Name: %s\nAuthor: %s\nQuantity: %d\n",
                   b.name, b.author, b.quantity);
            found = 1;
            break;
        }
    }
    if (!found)
```

```
        printf("Book not found!\n");

    fclose(fp);
}

/* Issue Book */
void issueBook()
{
    struct book b;
    int id;
    FILE *fp = fopen("library.dat", "rb+");

    printf("Enter Book ID to issue: ");
    scanf("%d", &id);

    while (fread(&b, sizeof(b), 1, fp))
    {
        if (b.id == id)
        {
            if (b.quantity > 0)
            {
                b.quantity--;
                fseek(fp, -sizeof(b), SEEK_CUR);
                fwrite(&b, sizeof(b), 1, fp);
                printf("Book issued successfully!\n");
            }
            else
            {
                printf("Book out of stock!\n");
            }
            fclose(fp);
            return;
        }
    }
    printf("Book not found!\n");
    fclose(fp);
}

/* Return Book */
void returnBook()
{
    struct book b;
    int id;
    FILE *fp = fopen("library.dat", "rb+");

    printf("Enter Book ID to return: ");
    scanf("%d", &id);

    while (fread(&b, sizeof(b), 1, fp))
    {
        if (b.id == id)
        {
            b.quantity++;
            fseek(fp, -sizeof(b), SEEK_CUR);
            fwrite(&b, sizeof(b), 1, fp);
            printf("Book returned successfully!\n");
            fclose(fp);
            return;
        }
    }
    printf("Book not found!\n");
    fclose(fp);
}
```

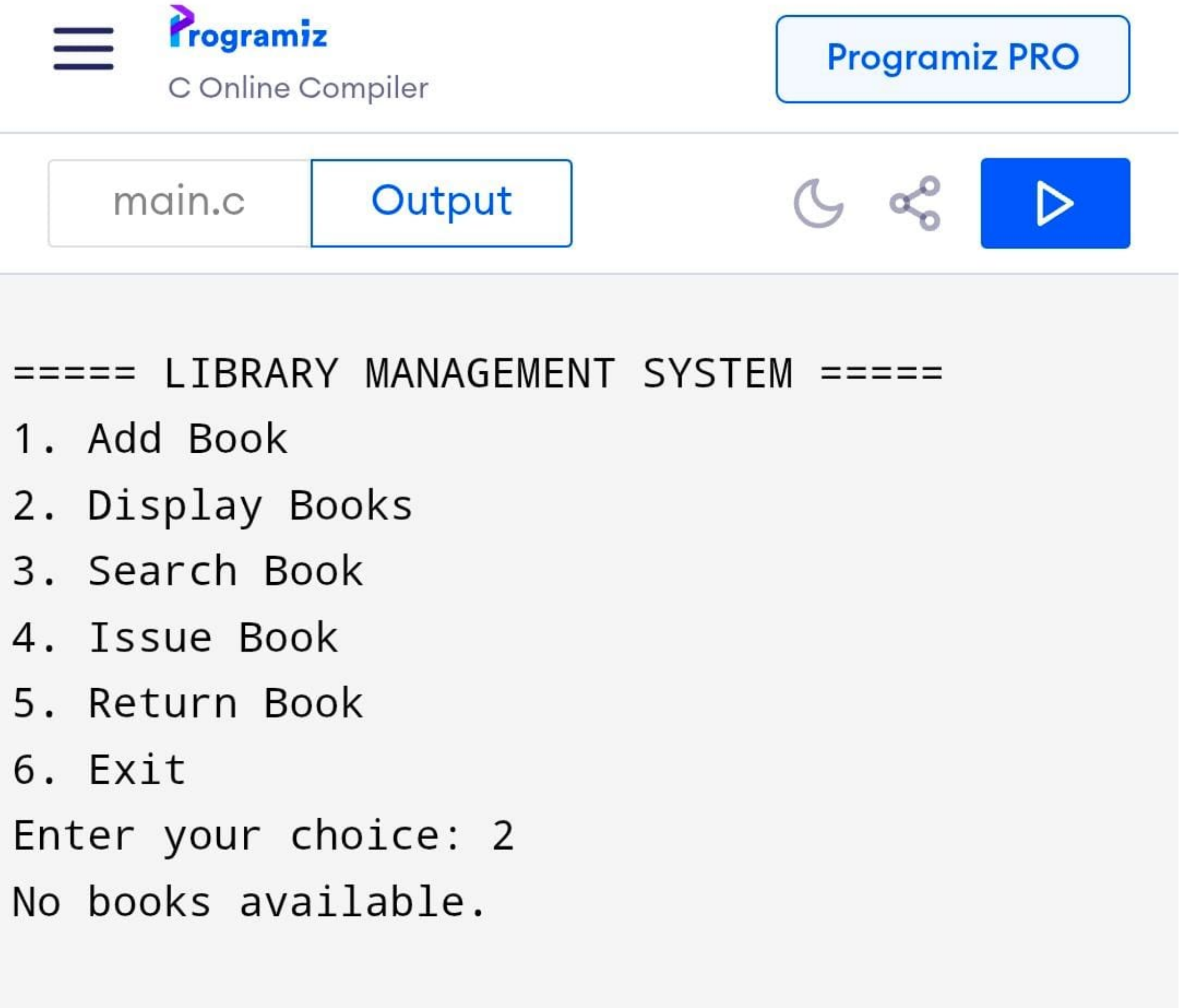
Result

Add Book



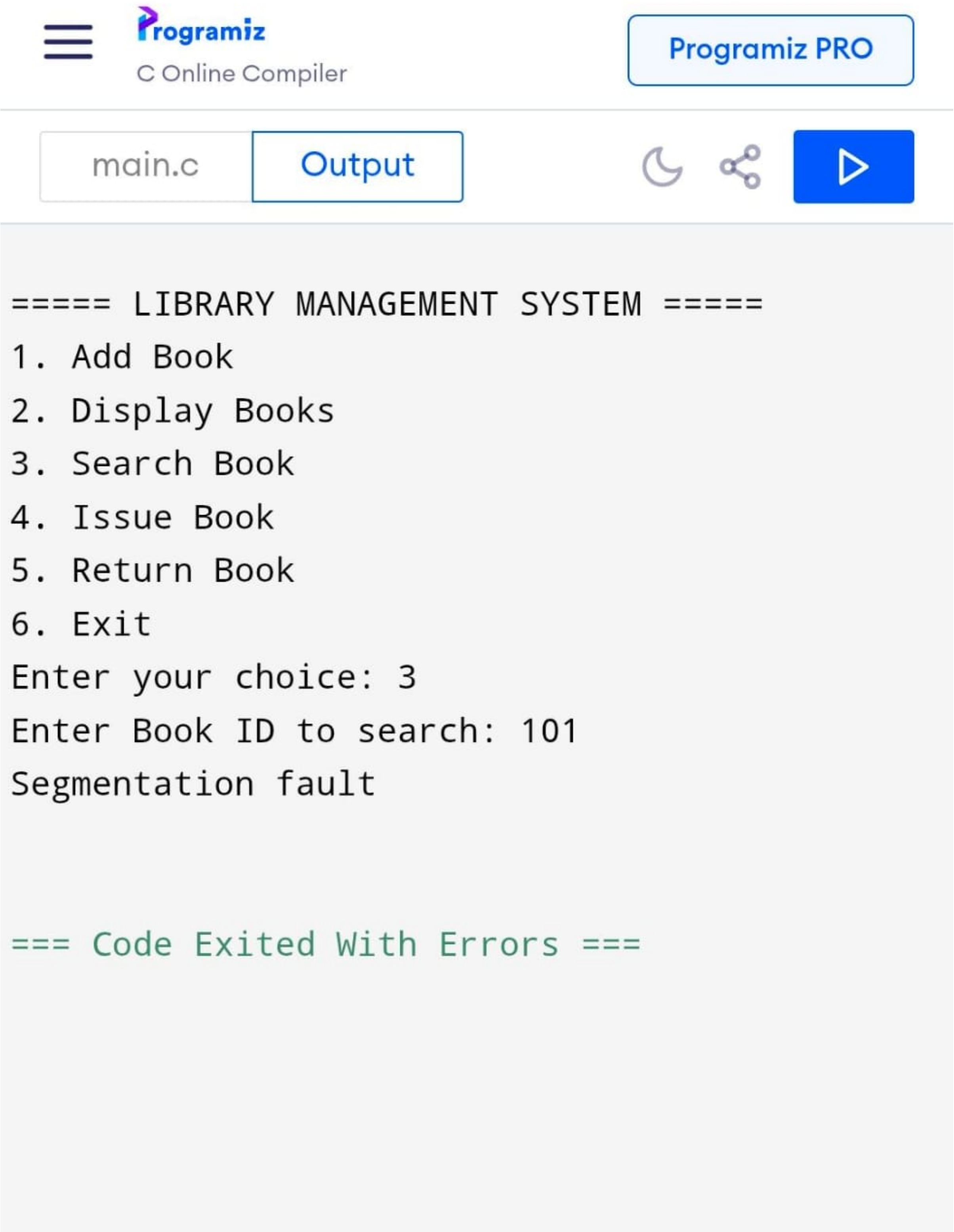
```
===== LIBRARY MANAGEMENT SYSTEM =====
1. Add Book
2. Display Books
3. Search Book
4. Issue Book
5. Return Book
6. Exit
Enter your choice: 1
Enter Book ID: 101
Enter Book Name: C programming
Enter Author Name: Dennis Ritchie
Enter Quantity: 5
Segmentation fault


=== Code Exited With Errors ===
```

Display Books



```
===== LIBRARY MANAGEMENT SYSTEM =====
1. Add Book
2. Display Books
3. Search Book
4. Issue Book
5. Return Book
6. Exit
Enter your choice: 2
No books available.
```

DOCMENTATION

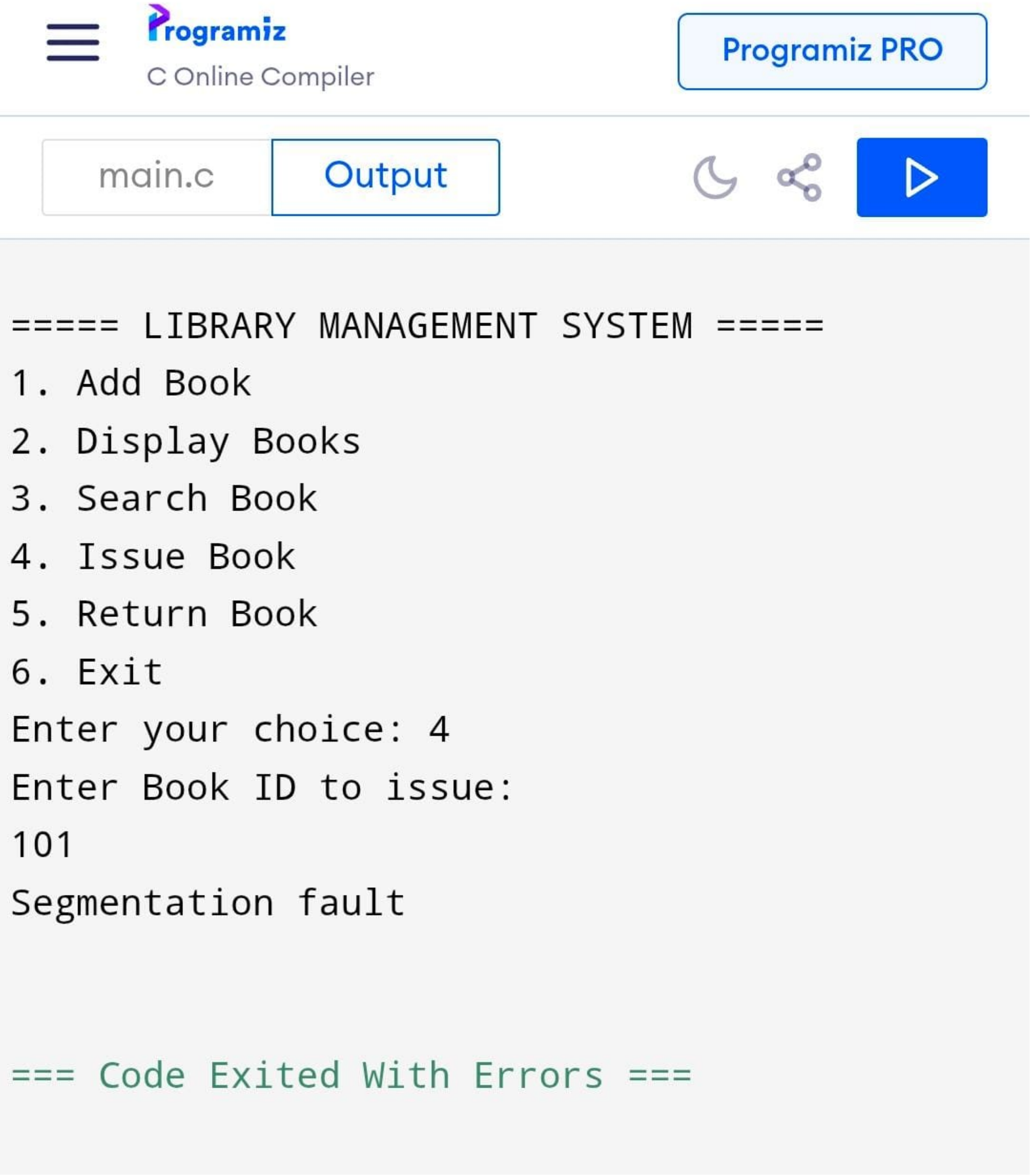


```
===== LIBRARY MANAGEMENT SYSTEM =====
1. Add Book
2. Display Books
3. Search Book
4. Issue Book
5. Return Book
6. Exit
Enter your choice: 3
Enter Book ID to search: 101
Segmentation fault


=== Code Exited With Errors ===
```

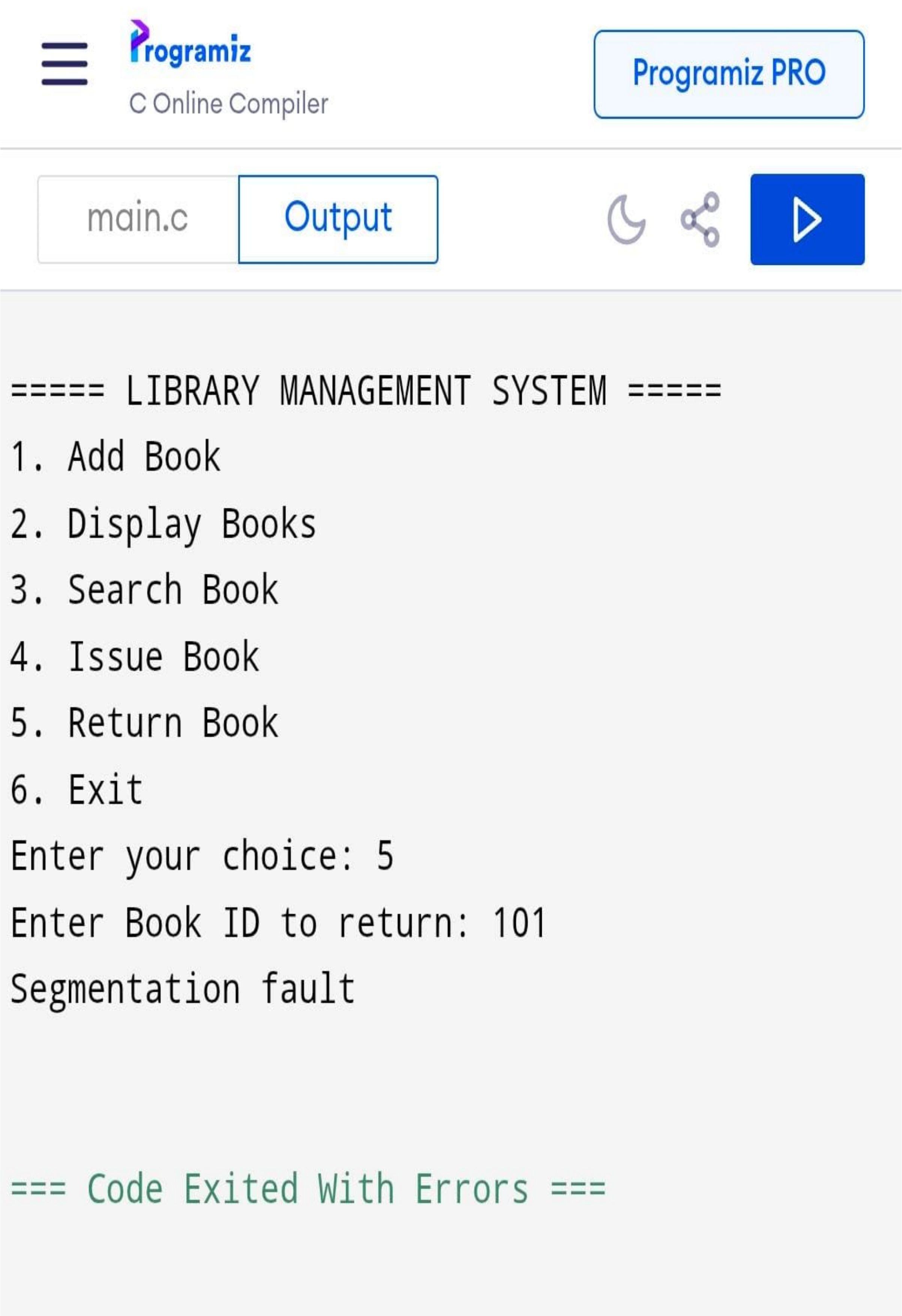
Search Book

DOCMENTATION

Issue Book



```
===== LIBRARY MANAGEMENT SYSTEM =====
1. Add Book
2. Display Books
3. Search Book
4. Issue Book
5. Return Book
6. Exit
Enter your choice: 4
Enter Book ID to issue:
101
Segmentation fault


=== Code Exited With Errors ===
```

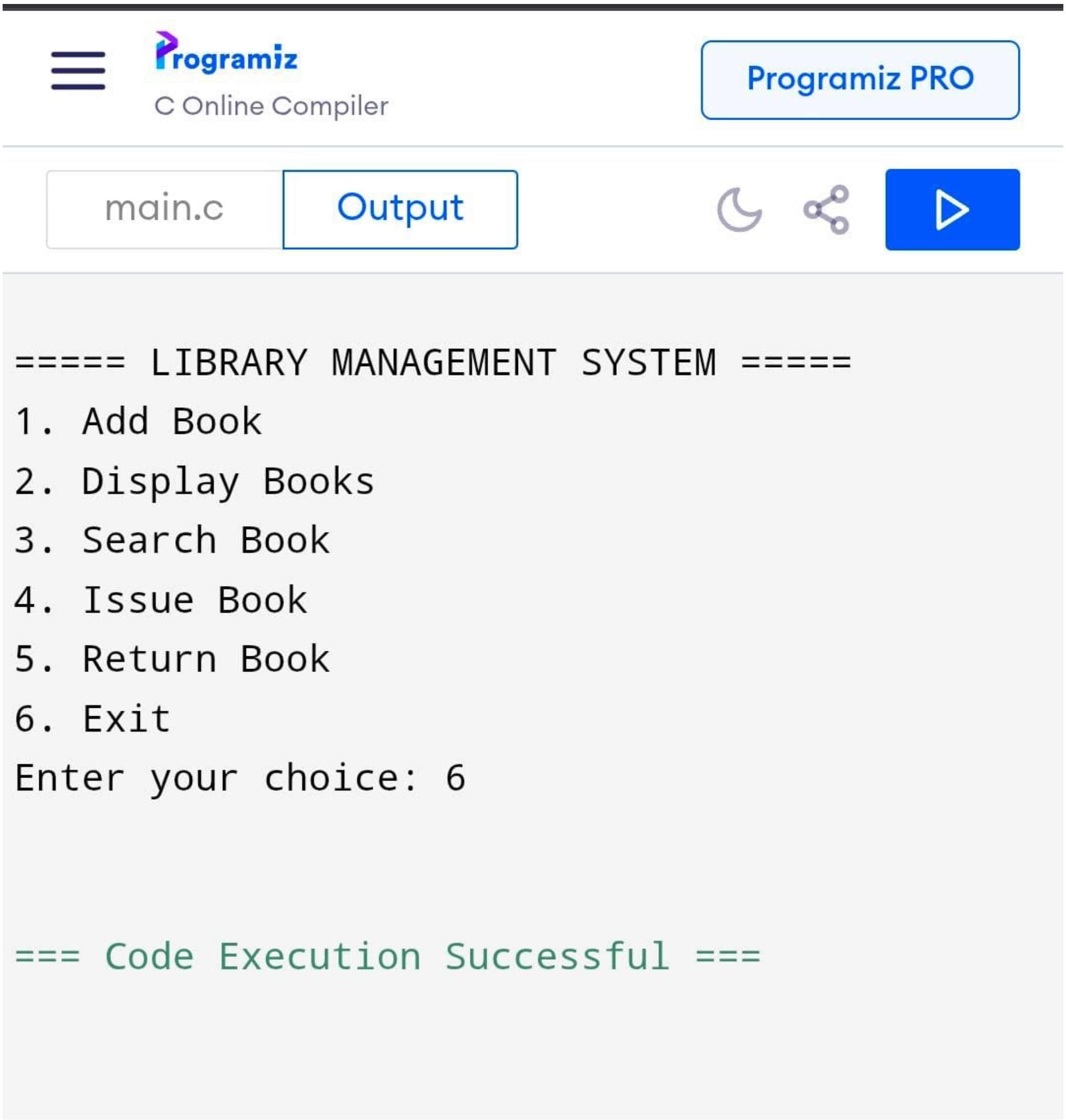


```
===== LIBRARY MANAGEMENT SYSTEM =====
1. Add Book
2. Display Books
3. Search Book
4. Issue Book
5. Return Book
6. Exit
Enter your choice: 5
Enter Book ID to return: 101
Segmentation fault


=== Code Exited With Errors ===
```

Return Book

DOCMENTATION

Exist



```
===== LIBRARY MANAGEMENT SYSTEM =====
1. Add Book
2. Display Books
3. Search Book
4. Issue Book
5. Return Book
6. Exit
Enter your choice: 6


=== Code Execution Successful ===
```

Conclusion

DOCMENTATION

The Library Management System provides an effective and reliable solution for managing library operations. It simplifies the process of maintaining records, reduces workload, and enhances accuracy. By automating library activities, the system saves time and ensures better organization of resources. Overall, the Library Management System improves the efficiency and quality of library services and supports the smooth functioning of modern libraries.