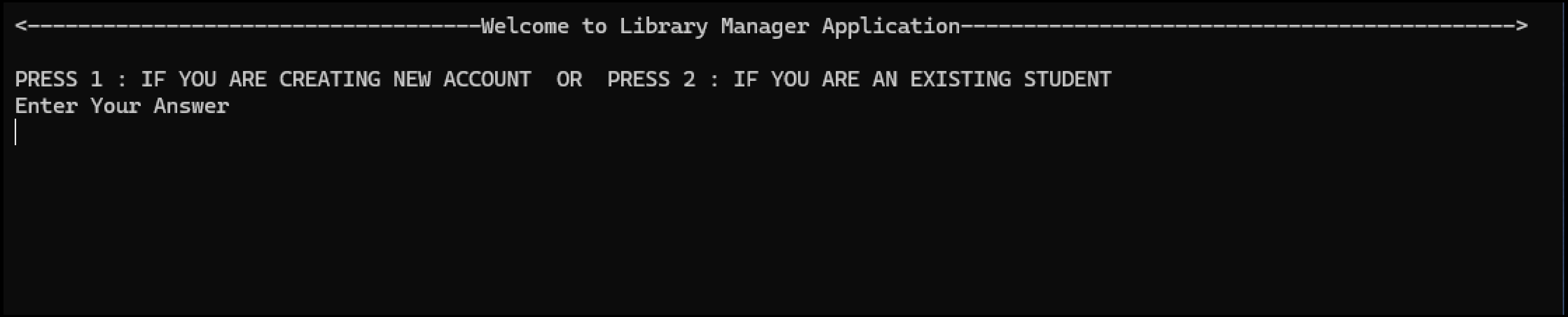
LIBRARY MANAGEMENT SYSTEM

USING C PROGRAMMING LANGUAGE

PURPOSE:

Provide an overview of the project, highlighting its significance in managing library records efficiently.



```
<------------------------------------Welcome to Library Manager Application-------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
```

SLIDE 2: WHY WE USED STRUCTURES

- **DATA ORGANIZATION:**

Structures enable us to group related data together, making it easier to manage and understand. For example, a student structure groups student ID and name.

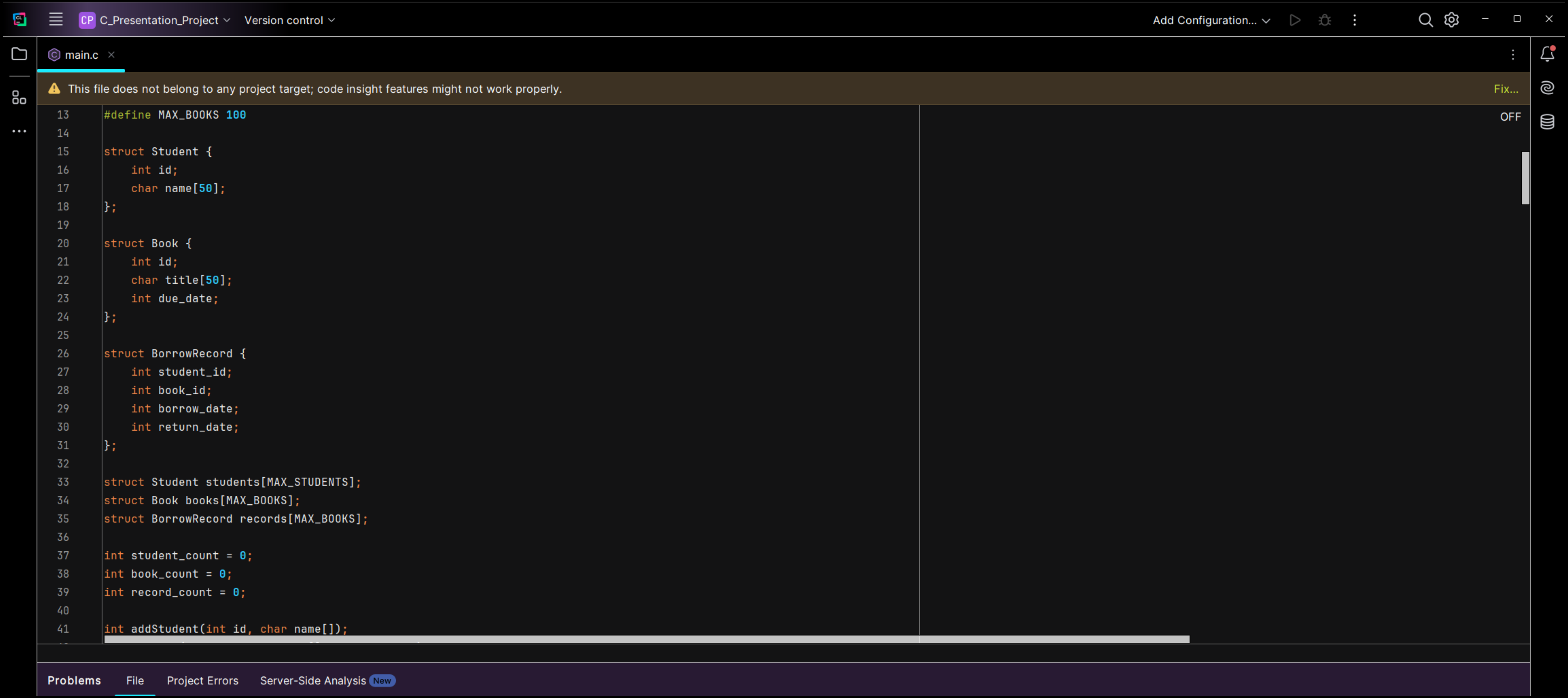
- **CODE READABILITY:**

Grouping related data into structures improves the readability and maintainability of the code.

- **HANDLING MULTIPLE DATA TYPES:**

Structures allow us to handle different data types under a single entity, such as strings and integers in the `Student` structure.

STRUCTURES USED IN THIS PROGRAM:



This file does not belong to any project target; code insight features might not work properly. Fix...

```c
#define MAX_BOOKS 100

struct Student {
    int id;
    char name[50];
};

struct Book {
    int id;
    char title[50];
    int due_date;
};

struct BorrowRecord {
    int student_id;
    int book_id;
    int borrow_date;
    int return_date;
};

struct Student students[MAX_STUDENTS];
struct Book books[MAX_BOOKS];
struct BorrowRecord records[MAX_BOOKS];

int student_count = 0;
int book_count = 0;
int record_count = 0;

int addStudent(int id, char name[]);
```

SLIDE 3: STRUCTURES USED

- **STUDENT STRUCTURE:**
- -ATTRIBUTES : `id` (integer), `name` (character array).
- -PURPOSE : To store and manage details of students.
- **BOOK STRUCTURE:**
- - ATTRIBUTES : `id` (integer), `title` (character array), `due_date` (integer).
- - PURPOSE : To store and manage details of books.
- **BORROWRECORD STRUCTURE:**
- - ATTRIBUTES : `student_id` (integer), `book_id` (integer), `borrow_date` (integer), `return_date` (integer).
- - PURPOSE : To keep a record of books borrowed by students, including borrow and return dates.

SLIDE 4: WHY WE USED FUNCTIONS

- **CODE REUSABILITY:**

  Functions allow us to reuse code, avoiding duplication and making the code more modular.

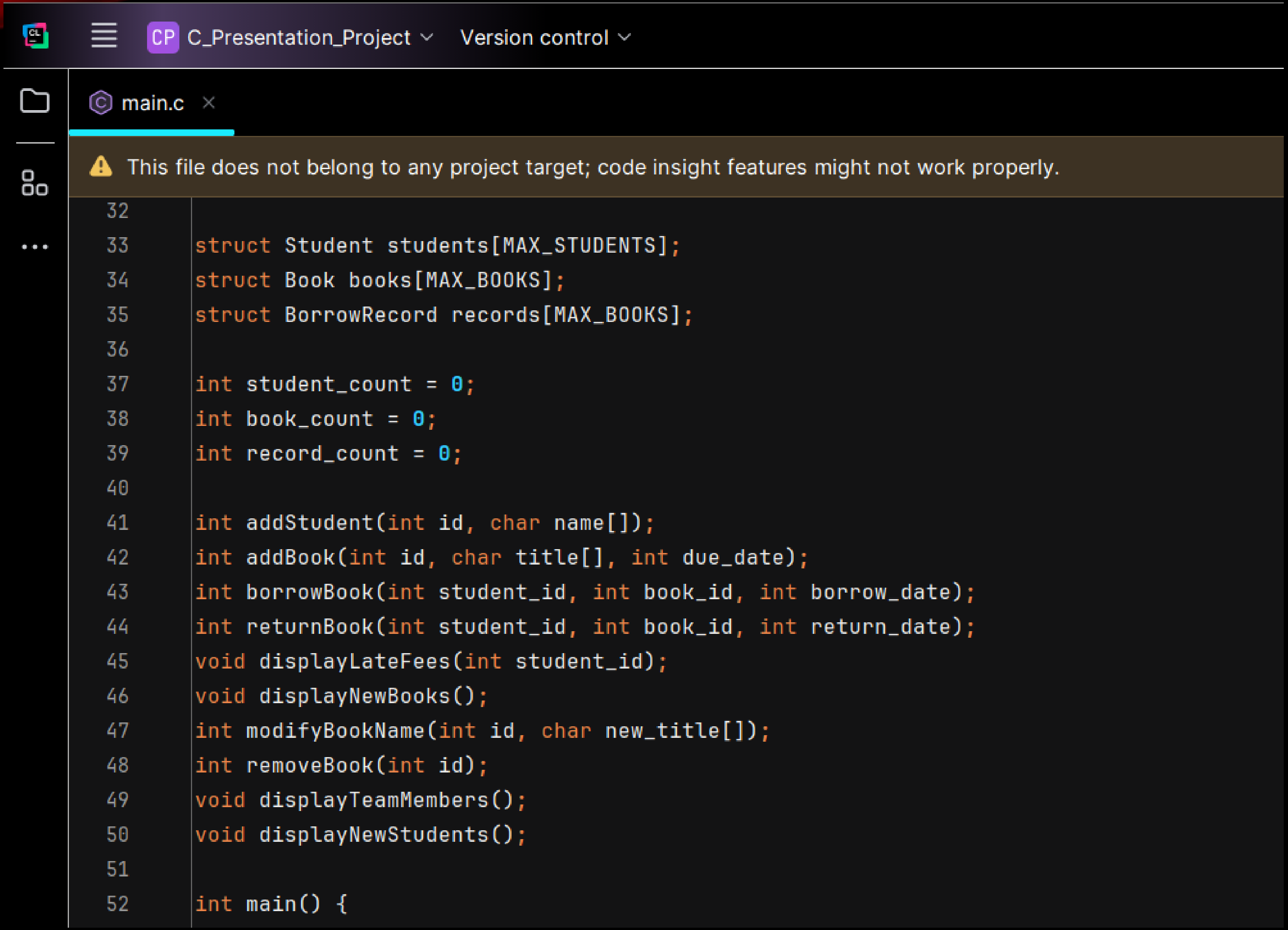
- **SIMPLIFIED CODE:**

  Breaking down the code into functions makes it easier to understand and manage.

- **EASE OF MAINTENANCE:**

  Functions make it easier to update and debug the code as changes can be made in a specific function without affecting others.

DECLARED FUNCTIONS :



This file does not belong to any project target; code insight features might not work properly.

```c

struct Student students[MAX_STUDENTS];
struct Book books[MAX_BOOKS];
struct BorrowRecord records[MAX_BOOKS];

int student_count = 0;
int book_count = 0;
int record_count = 0;

int addStudent(int id, char name[]);
int addBook(int id, char title[], int due_date);
int borrowBook(int student_id, int book_id, int borrow_date);
int returnBook(int student_id, int book_id, int return_date);
void displayLateFees(int student_id);
void displayNewBooks();
int modifyBookName(int id, char new_title[]);
int removeBook(int id);
void displayTeamMembers();
void displayNewStudents();

int main() {
```

SLIDE 5: FUNCTIONS USED

1. addStudent: Adds a new student to the system.
2. addBook: Adds a new book to the library's inventory.
3. borrowBook: Records the borrowing of a book by a student.
4. returnBook: Records the return of a book by a student.
5. displayLateFees:Calculates and displays late fees for a student.
6. modifyBookName:Modifies the title of an existing book.
7. removeBook: Removes a book from the library's inventory.
8. displayTeamMembers: Displays the names of the team members.
9. displayNewStudents: Displays a list of new students.

SLIDE 6: HOW WE USED FUNCTIONS

- ❖ USER INTERACTION: Functions are called based on user inputs, ensuring that the system responds to user commands efficiently.
- ❖ DATA MANIPULATION: Each function manipulates the respective data structures to perform its task, such as adding a new book or updating a student's record.
- ❖ VALIDATION: Functions include validation checks to ensure data integrity and handle edge cases, such as checking for maximum capacity of students or books.

SLIDE 7: FINDING THE LOGIC

- UNDERSTANDING REQUIREMENTS:

  We started by understanding the user requirements, defining what the system needs to do (e.g., add students, borrow books).

- BREAKING DOWN TASKS:

  We divided the problem into smaller, manageable tasks (e.g., creating functions for each operation).

- FLOW CONTROL:

  We used decision-making statements like `if-else` to control the flow of the program based on user inputs.

SLIDE 8: IMPLEMENTING THE LOGIC

- USER INPUT HANDLING:

We used functions like `scanf` and `fgets` to get input from the user and store it in appropriate variables.

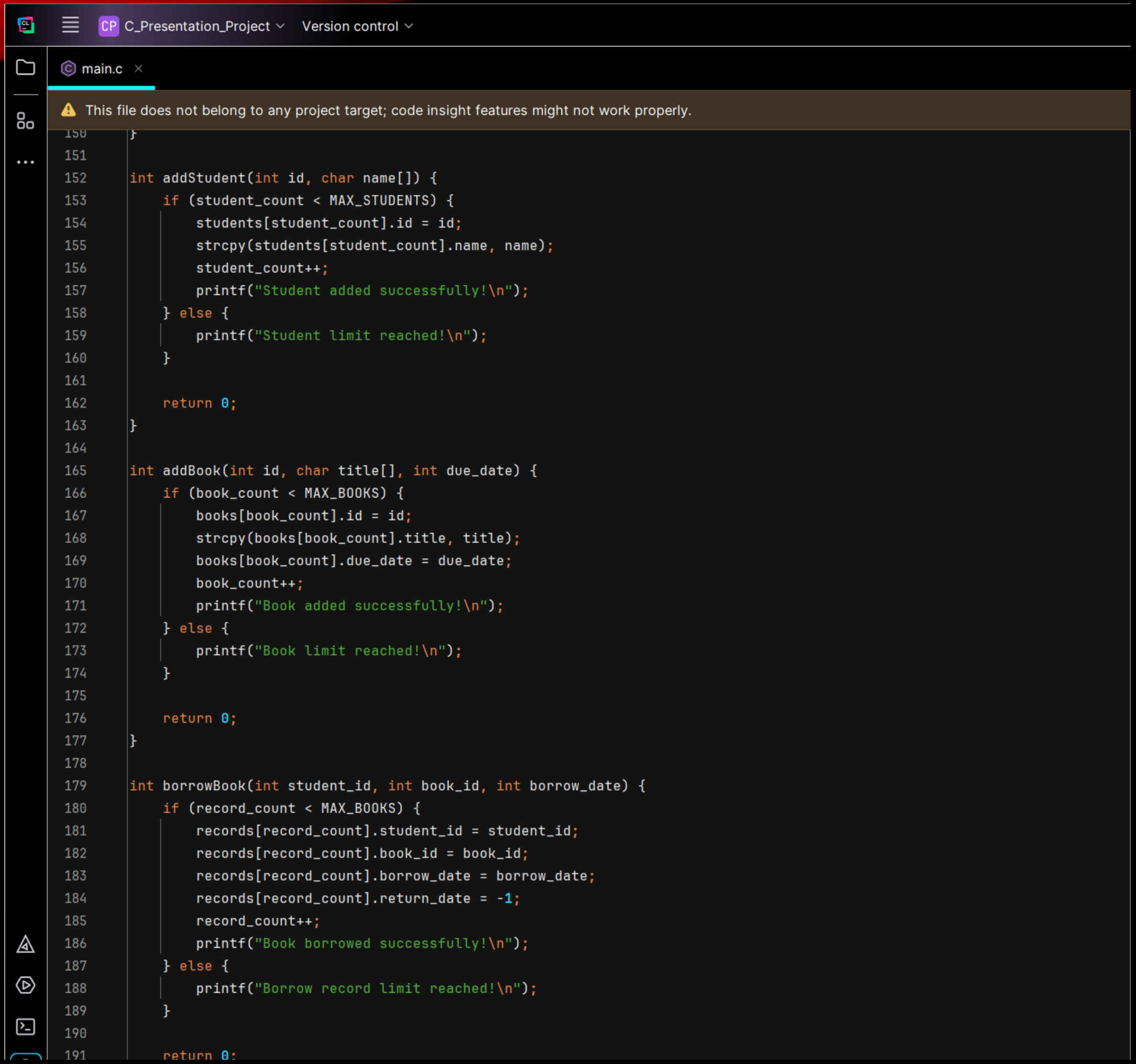
- DATA STORAGE:

We stored data in arrays of structures, enabling efficient management and retrieval of information.

- CONDITIONS AND LOOPS:

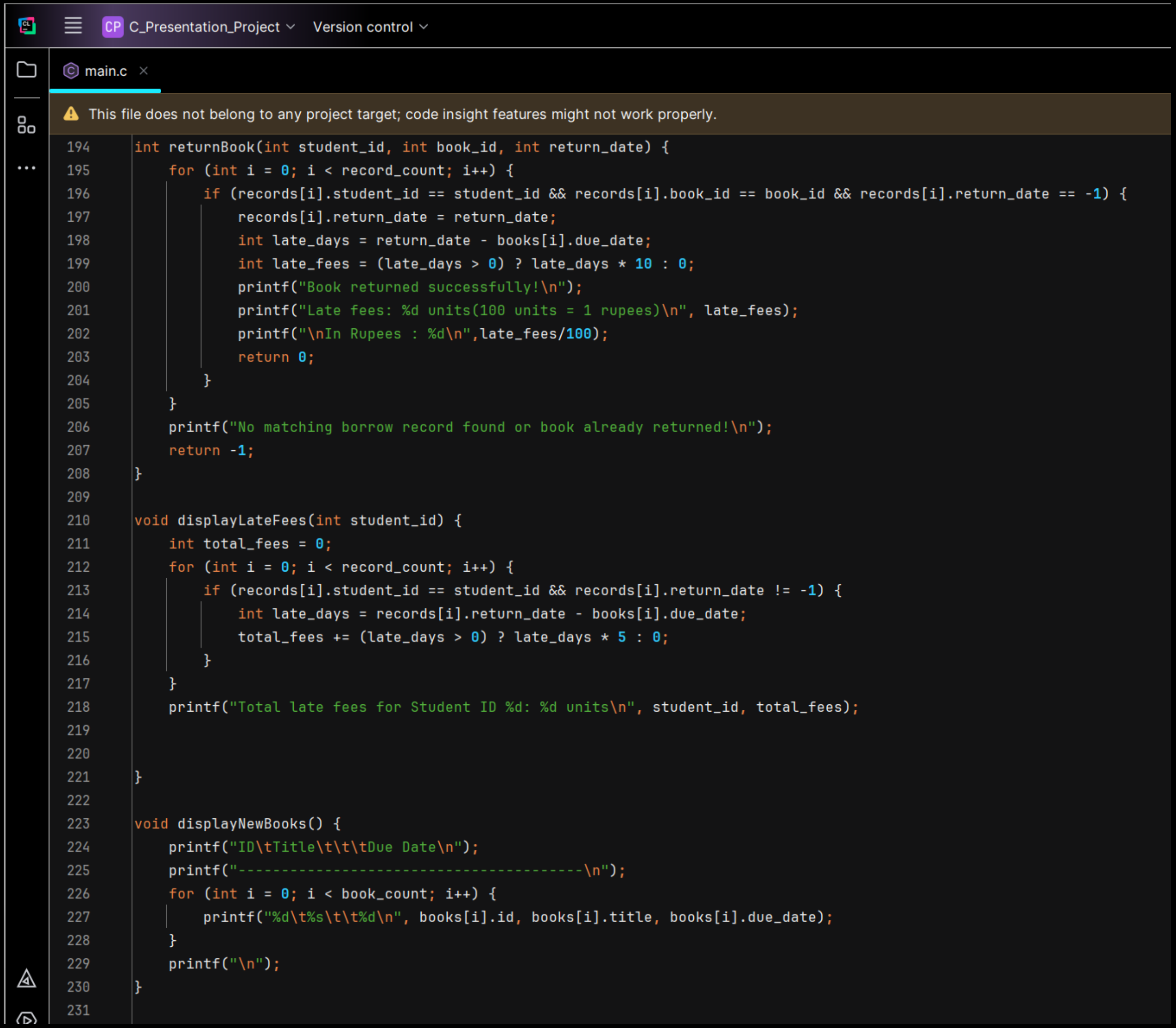
We used conditions to validate data and loops to iterate through arrays, ensuring that operations like adding students or books are performed correctly.

LOGIC USED IN THE PROGRAM:



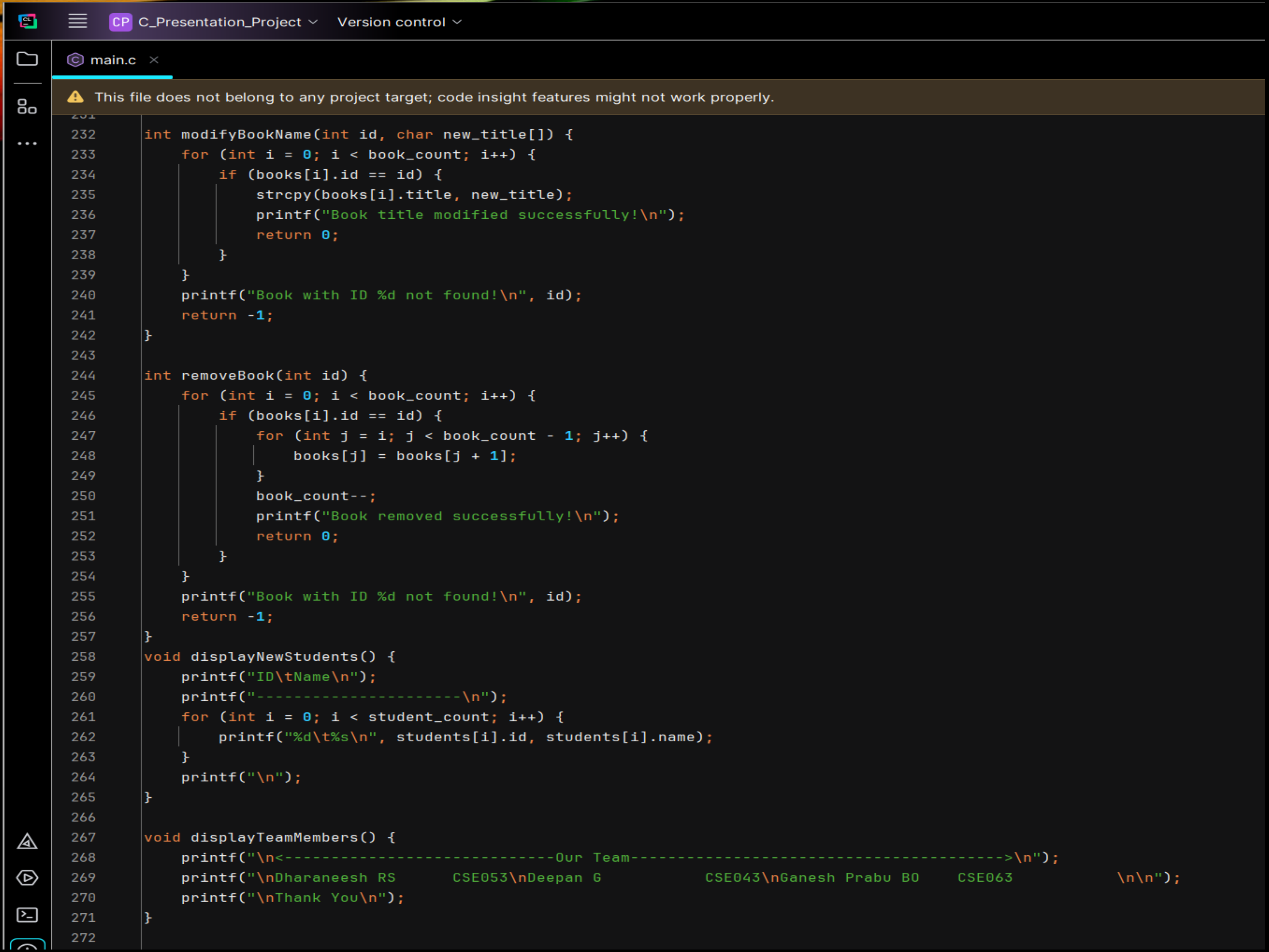
This file does not belong to any project target; code insight features might not work properly.

```c
}

int addStudent(int id, char name[]) {
    if (student_count < MAX_STUDENTS) {
        students[student_count].id = id;
        strcpy(students[student_count].name, name);
        student_count++;
        printf("Student added successfully!\n");
    } else {
        printf("Student limit reached!\n");
    }

    return 0;
}

int addBook(int id, char title[], int due_date) {
    if (book_count < MAX_BOOKS) {
        books[book_count].id = id;
        strcpy(books[book_count].title, title);
        books[book_count].due_date = due_date;
        book_count++;
        printf("Book added successfully!\n");
    } else {
        printf("Book limit reached!\n");
    }

    return 0;
}

int borrowBook(int student_id, int book_id, int borrow_date) {
    if (record_count < MAX_BOOKS) {
        records[record_count].student_id = student_id;
        records[record_count].book_id = book_id;
        records[record_count].borrow_date = borrow_date;
        records[record_count].return_date = -1;
        record_count++;
        printf("Book borrowed successfully!\n");
    } else {
        printf("Borrow record limit reached!\n");
    }

    return 0;
```



This file does not belong to any project target; code insight features might not work properly.

```c
int returnBook(int student_id, int book_id, int return_date) {
    for (int i = 0; i < record_count; i++) {
        if (records[i].student_id == student_id && records[i].book_id == book_id && records[i].return_date == -1) {
            records[i].return_date = return_date;
            int late_days = return_date - books[i].due_date;
            int late_fees = (late_days > 0) ? late_days * 10 : 0;
            printf("Book returned successfully!\n");
            printf("Late fees: %d units(100 units = 1 rupees)\n", late_fees);
            printf("\nIn Rupees : %d\n",late_fees/100);
            return 0;
        }
    }
    printf("No matching borrow record found or book already returned!\n");
    return -1;
}

void displayLateFees(int student_id) {
    int total_fees = 0;
    for (int i = 0; i < record_count; i++) {
        if (records[i].student_id == student_id && records[i].return_date != -1) {
            int late_days = records[i].return_date - books[i].due_date;
            total_fees += (late_days > 0) ? late_days * 5 : 0;
        }
    }
    printf("Total late fees for Student ID %d: %d units\n", student_id, total_fees);


}

void displayNewBooks() {
    printf("ID\tTitle\t\t\tDue Date\n");
    printf("----------------------------------------\n");
    for (int i = 0; i < book_count; i++) {
        printf("%d\t%s\t\t%d\n", books[i].id, books[i].title, books[i].due_date);
    }
    printf("\n");
}

```



This file does not belong to any project target; code insight features might not work properly.

```c
int modifyBookName(int id, char new_title[]) {
    for (int i = 0; i < book_count; i++) {
        if (books[i].id == id) {
            strcpy(books[i].title, new_title);
            printf("Book title modified successfully!\n");
            return 0;
        }
    }
    printf("Book with ID %d not found!\n", id);
    return -1;
}

int removeBook(int id) {
    for (int i = 0; i < book_count; i++) {
        if (books[i].id == id) {
            for (int j = i; j < book_count - 1; j++) {
                books[j] = books[j + 1];
            }
            book_count--;
            printf("Book removed successfully!\n");
            return 0;
        }
    }
    printf("Book with ID %d not found!\n", id);
    return -1;
}
void displayNewStudents() {
    printf("ID\tName\n");
    printf("-------------------\n");
    for (int i = 0; i < student_count; i++) {
        printf("%d\t%s\n", students[i].id, students[i].name);
    }
    printf("\n");
}

void displayTeamMembers() {
    printf("\n<---------------------------Our Team------------------------------------->\n");
    printf("\nDharaneesh RS      CSE053\nDeepan G          CSE043\nGanesh Prabu BO    CSE063          \n\n");
    printf("\nThank You\n");
}

```

SLIDE 9: CALCULATING TEST CASES

- ADDING STUDENTS AND BOOKS:

  Tested with various inputs to ensure the system correctly adds new entries and handles edge cases (e.g., maximum capacity).

- BORROW AND RETURN BOOKS:

  Validated the borrowing and returning process with different scenarios to ensure accuracy.

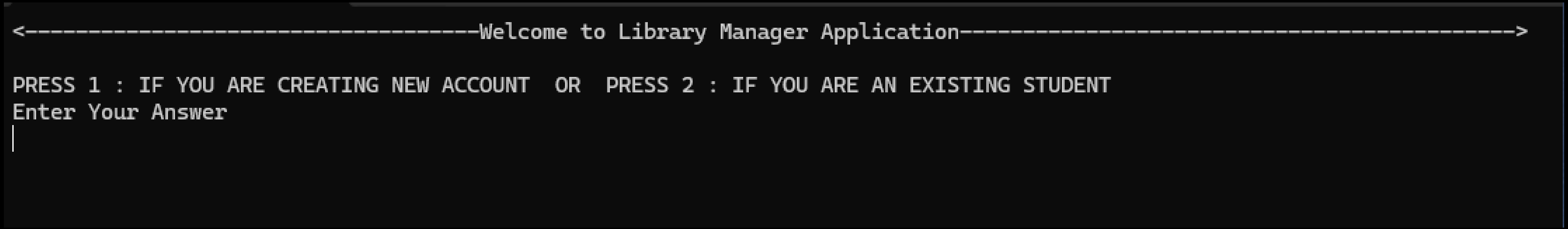
- MODIFY AND REMOVE BOOKS:

  Tested the modification and removal of books to ensure that the system updates or deletes records correctly.

- LATE FEES:

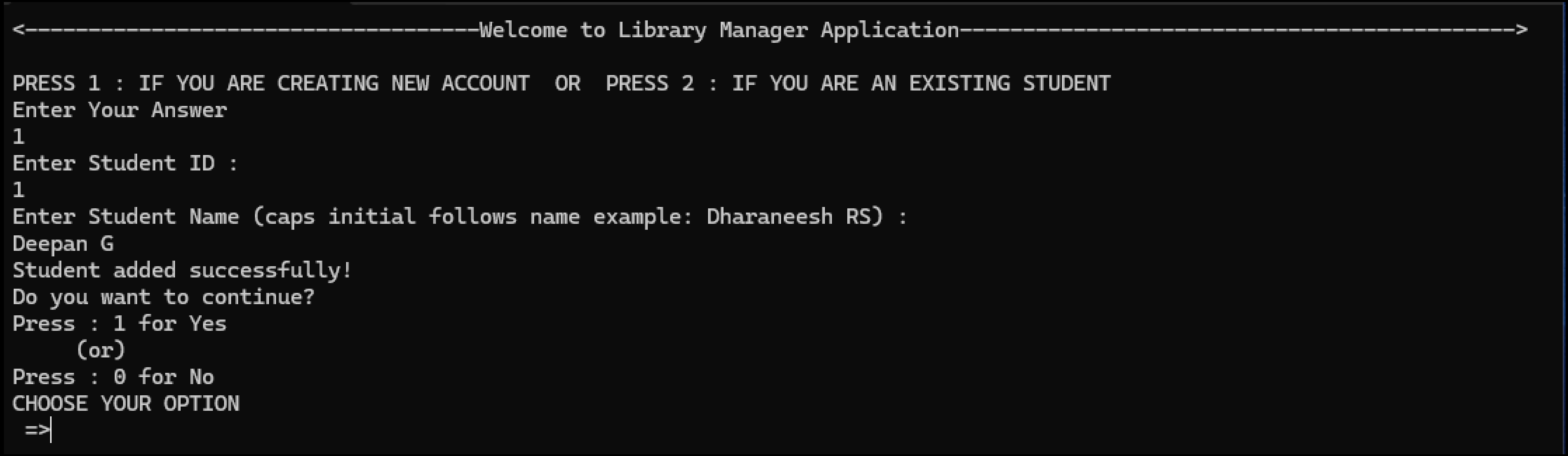
  Calculated late fees for different return dates to ensure the logic handles various cases accurately.

OUTPUT:



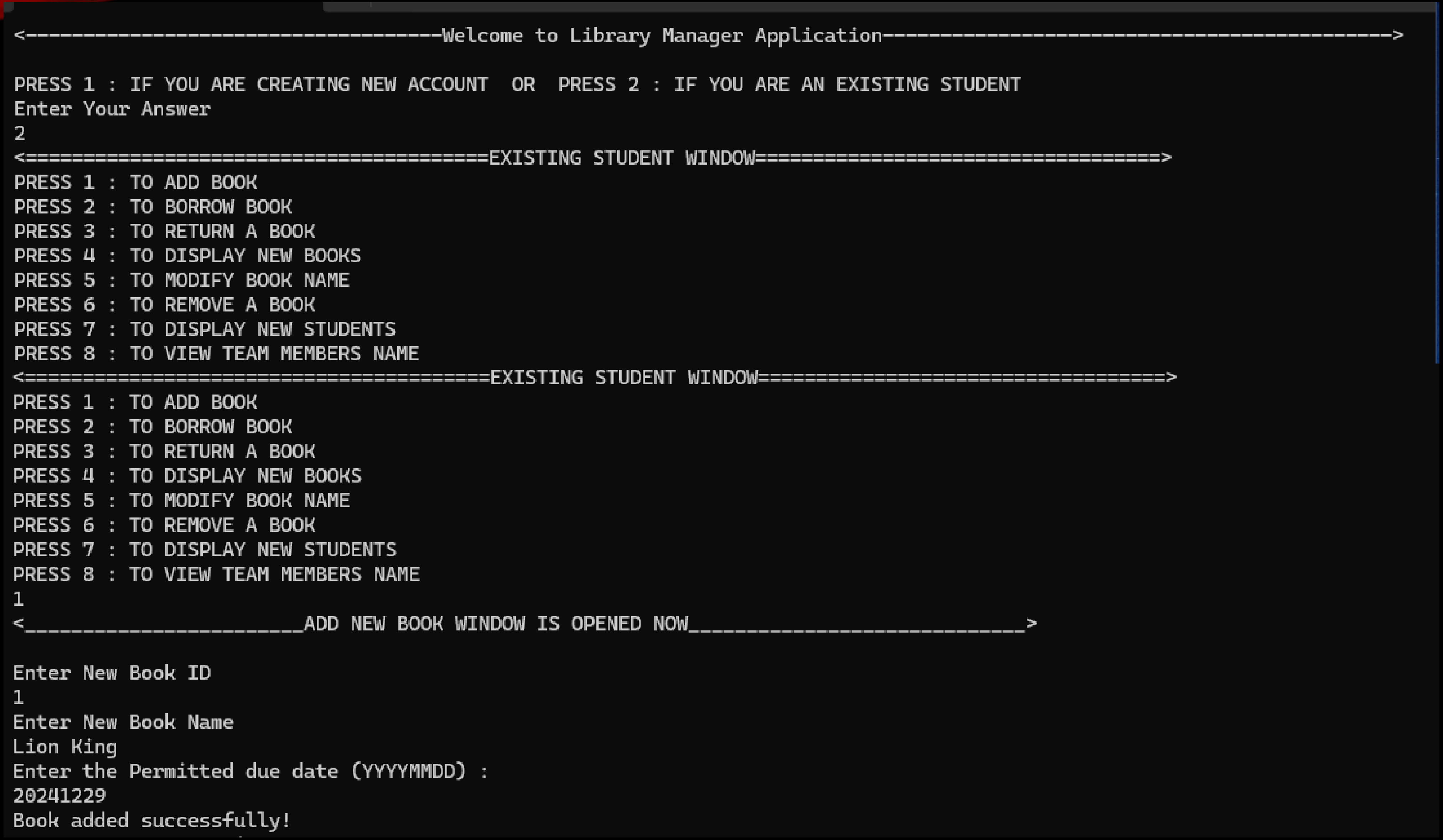
```
<------------------------------------Welcome to Library Manager Application-------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
```

- *ENTRY FOR NEW STUDENTS MENU:*



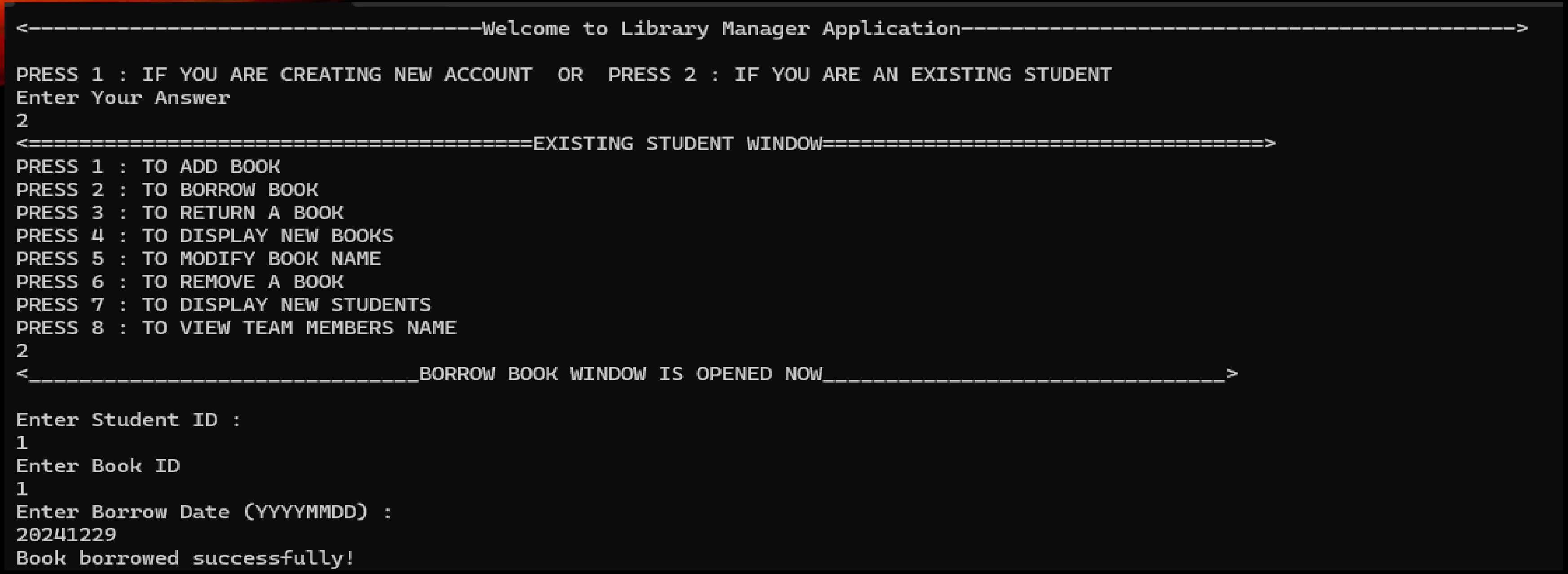
```
<------------------------------------Welcome to Library Manager Application-------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
1
Enter Student ID :
1
Enter Student Name (caps initial follows name example: Dharaneesh RS) :
Deepan G
Student added successfully!
Do you want to continue?
Press : 1 for Yes
     (or)
Press : 0 for No
CHOOSE YOUR OPTION
 =>
```

MENU'S FOR EXISTING STUDENT AND ADD NEW STUDENT:

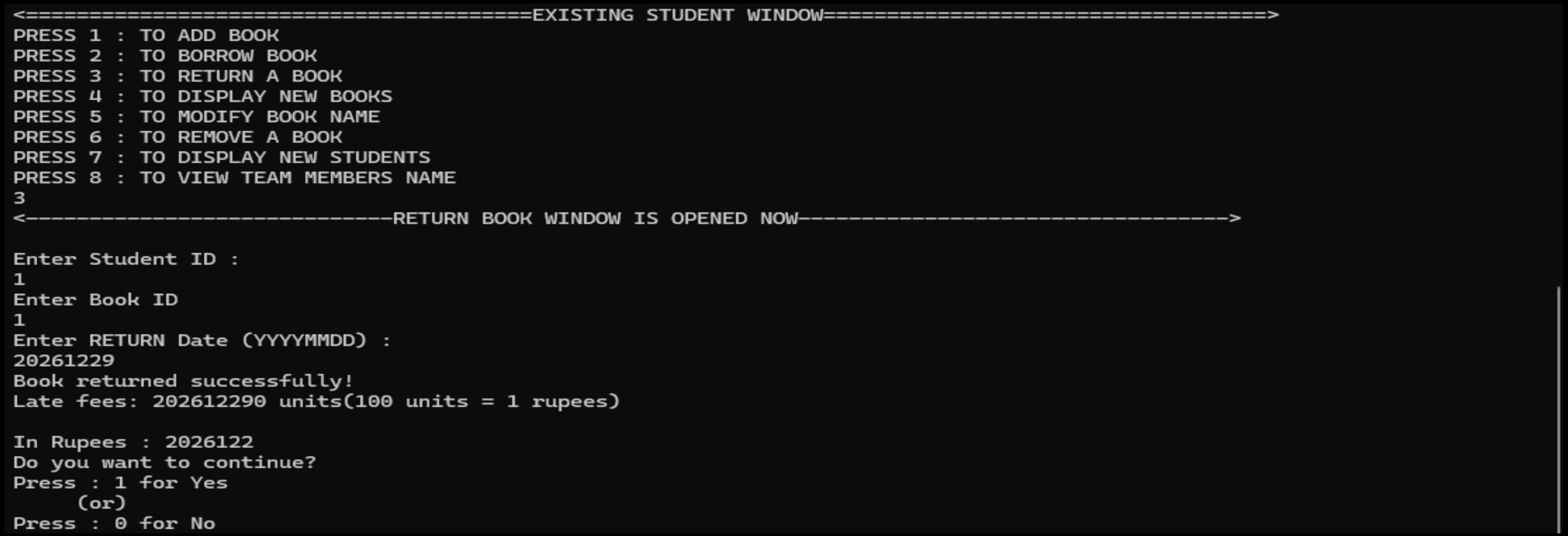


```
<------------------------------------Welcome to Library Manager Application------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
2
<========================================EXISTING STUDENT WINDOW====================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
<========================================EXISTING STUDENT WINDOW====================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
1
<________________________ADD NEW BOOK WINDOW IS OPENED NOW_____________________________>

Enter New Book ID
1
Enter New Book Name
Lion King
Enter the Permitted due date (YYYYMMDD) :
20241229
Book added successfully!
```

BORROW AND RETURN BOOK:

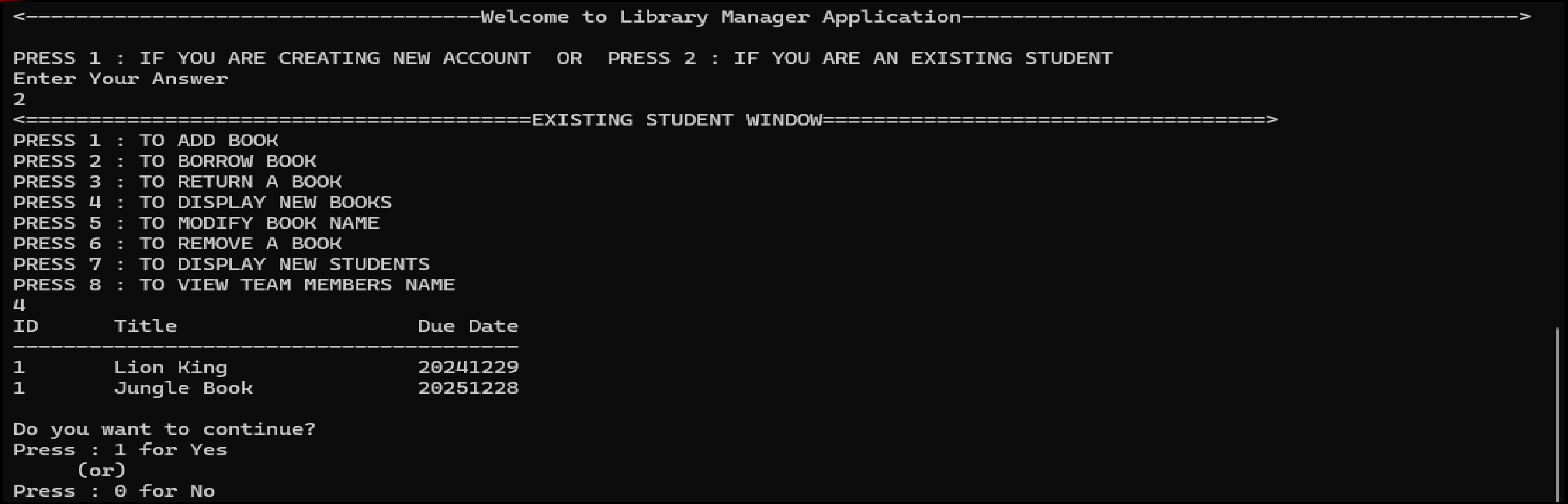


```
<------------------------------------Welcome to Library Manager Application-------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
2
<========================================EXISTING STUDENT WINDOW===================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
2
<_______________________________BORROW BOOK WINDOW IS OPENED NOW________________________________>

Enter Student ID :
1
Enter Book ID
1
Enter Borrow Date (YYYYMMDD) :
20241229
Book borrowed successfully!
```



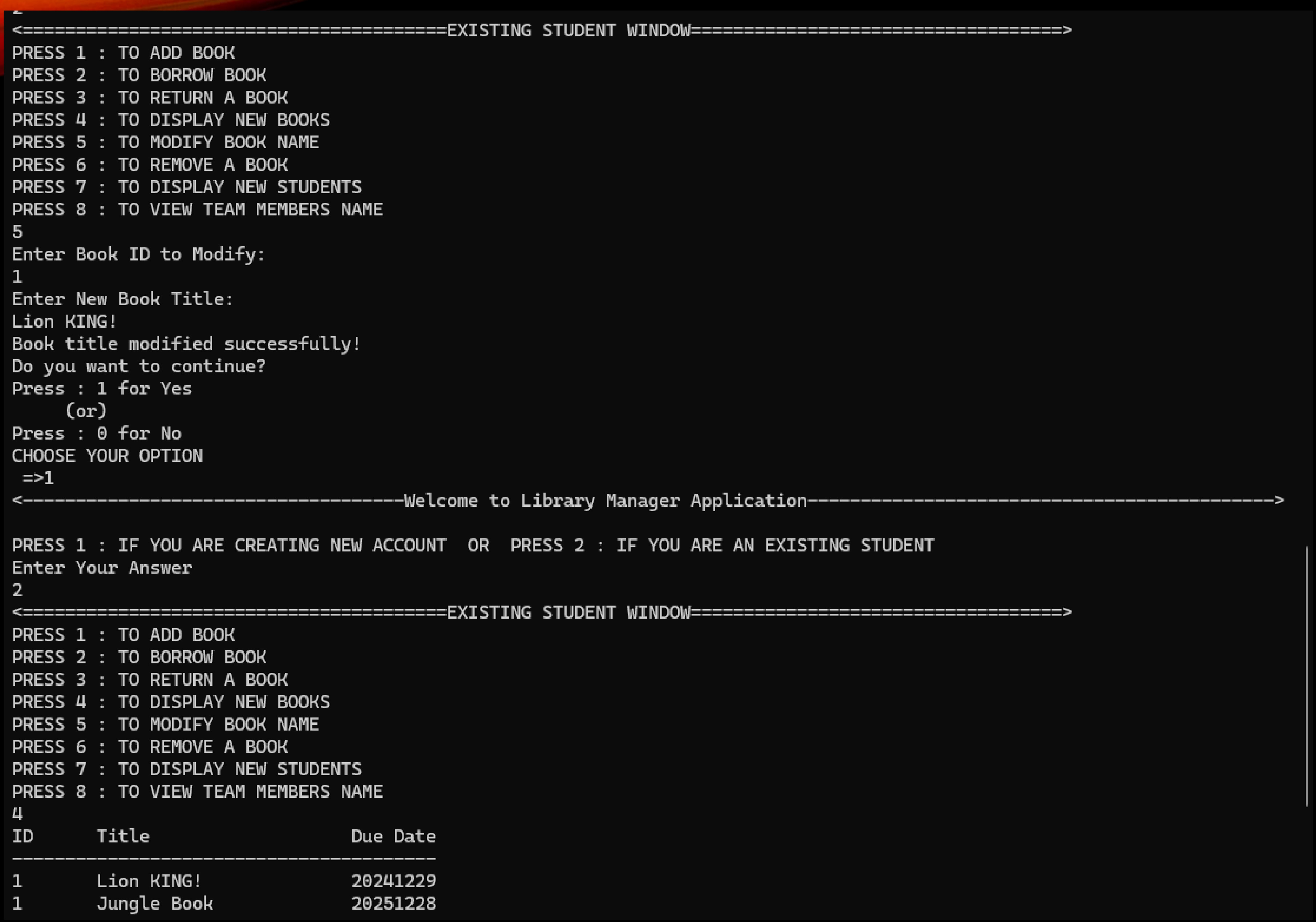
```
<========================================EXISTING STUDENT WINDOW===================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
3
<-----------------------------RETURN BOOK WINDOW IS OPENED NOW---------------------------------->

Enter Student ID :
1
Enter Book ID
1
Enter RETURN Date (YYYYMMDD) :
20261229
Book returned successfully!
Late fees: 202612290 units(100 units = 1 rupees)

In Rupees : 2026122
Do you want to continue?
Press : 1 for Yes
     (or)
Press : 0 for No
```

DISPLAY THE ENTRY OF NEW BOOK:



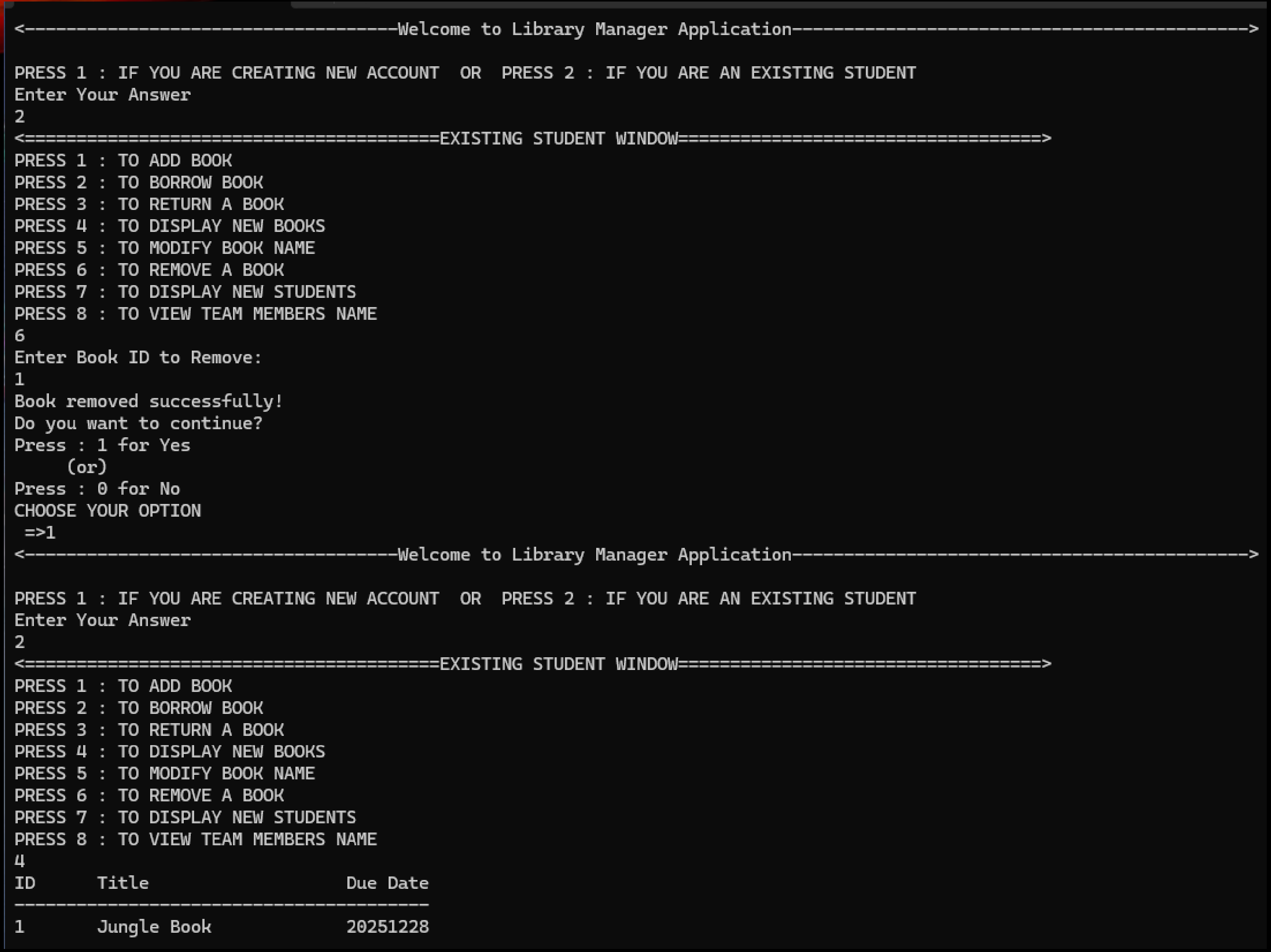
```
<------------------------------------Welcome to Library Manager Application-------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
2
<========================================EXISTING STUDENT WINDOW====================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
4
ID      Title                   Due Date
----------------------------------------
1       Lion King               20241229
1       Jungle Book             20251228

Do you want to continue?
Press : 1 for Yes
     (or)
Press : 0 for No
```

MODIFY THE ENTRY OF NEW BOOK:



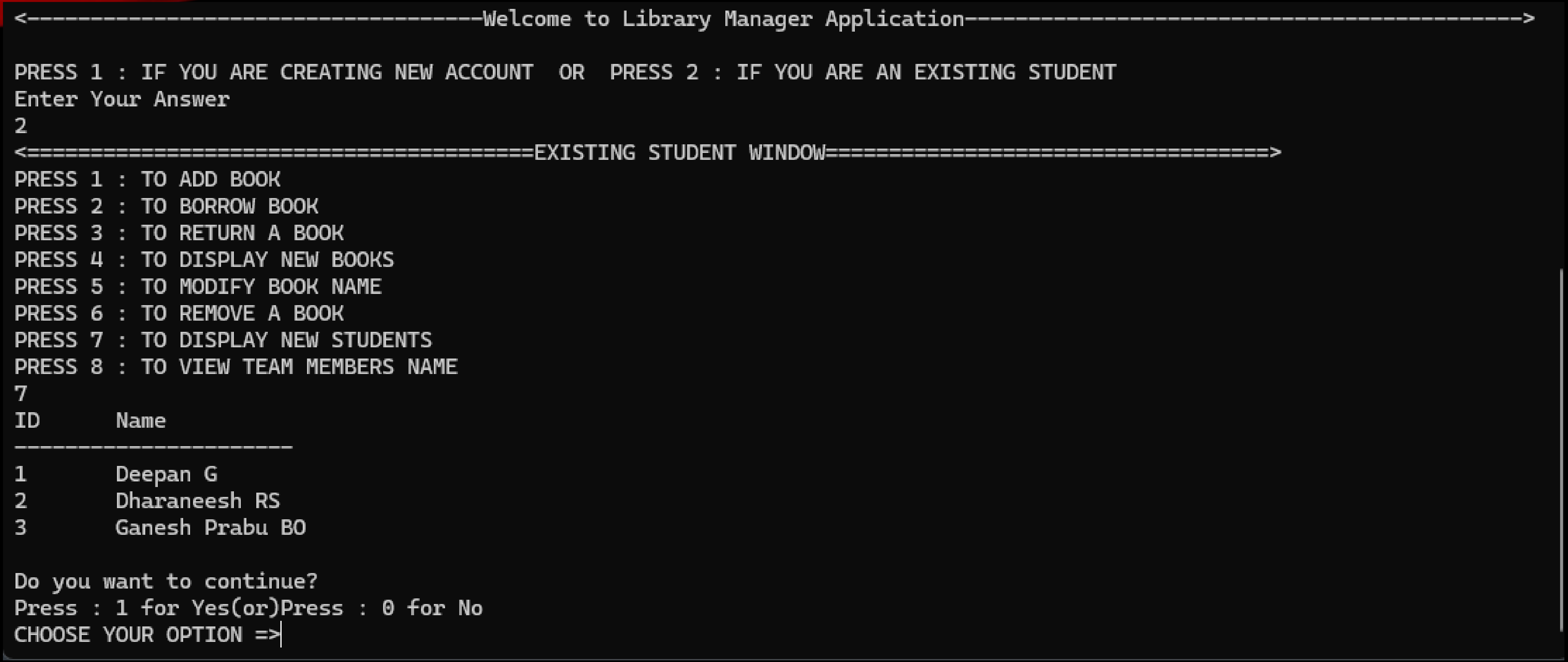
```
2
<========================================EXISTING STUDENT WINDOW===================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
5
Enter Book ID to Modify:
1
Enter New Book Title:
Lion KING!
Book title modified successfully!
Do you want to continue?
Press : 1 for Yes
     (or)
Press : 0 for No
CHOOSE YOUR OPTION
 =>1
<------------------------------------Welcome to Library Manager Application-------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
2
<========================================EXISTING STUDENT WINDOW===================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
4
ID      Title                   Due Date
----------------------------------------
1       Lion KING!              20241229
1       Jungle Book             20251228
```

REMOVE THE ENTRY OF NEW BOOK:



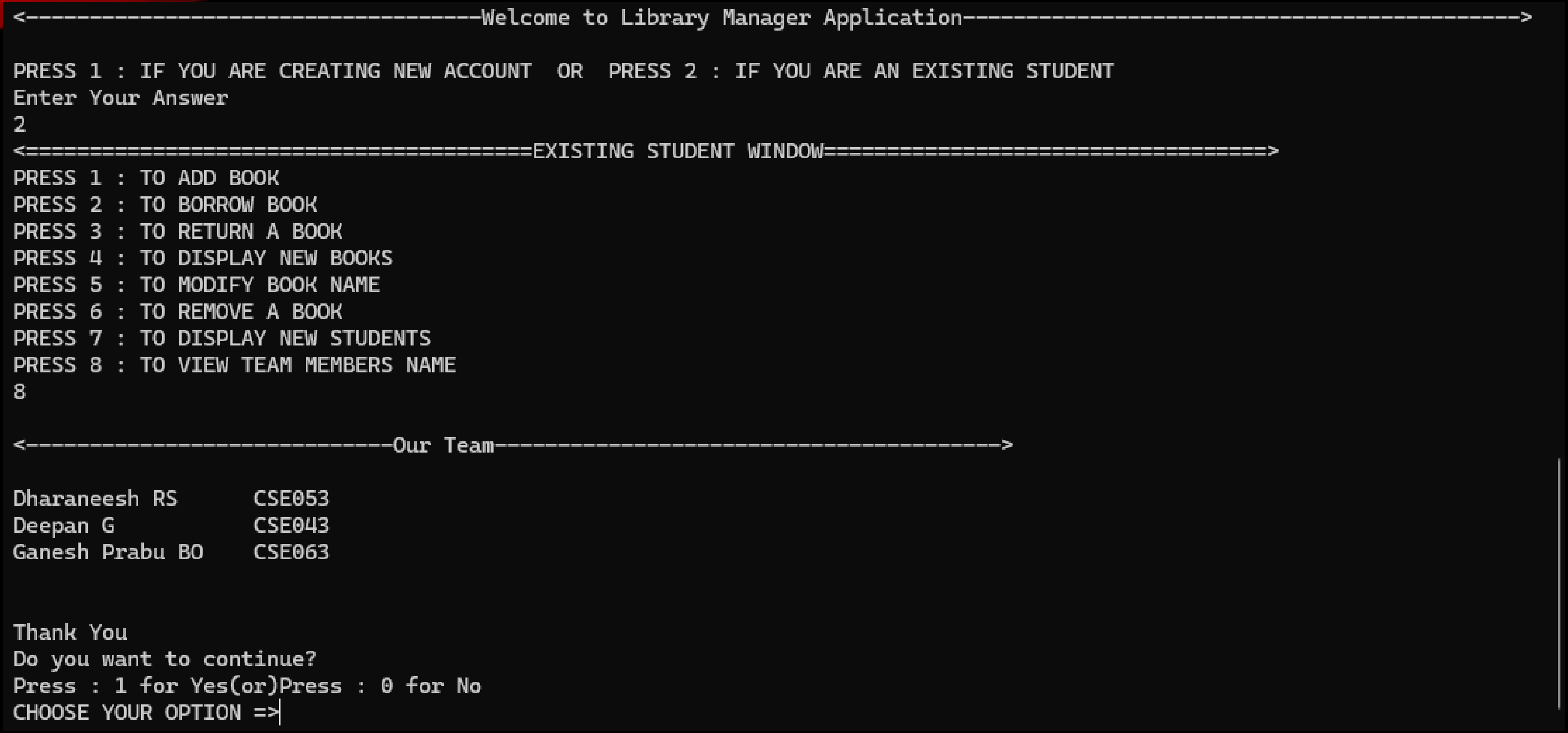
```
<------------------------------------Welcome to Library Manager Application-------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
2
<========================================EXISTING STUDENT WINDOW===================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
6
Enter Book ID to Remove:
1
Book removed successfully!
Do you want to continue?
Press : 1 for Yes
     (or)
Press : 0 for No
CHOOSE YOUR OPTION
 =>1
<------------------------------------Welcome to Library Manager Application-------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
2
<========================================EXISTING STUDENT WINDOW===================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
4
ID      Title                   Due Date
----------------------------------------
1       Jungle Book             20251228
```

DISPLAYING NEW STUDENTS NAME :



```
<------------------------------------Welcome to Library Manager Application-------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
2
<========================================EXISTING STUDENT WINDOW===================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
7
ID      Name
----------------------
1       Deepan G
2       Dharaneesh RS
3       Ganesh Prabu BO

Do you want to continue?
Press : 1 for Yes(or)Press : 0 for No
CHOOSE YOUR OPTION =>
```

TEAM MEMBERS NAME:



```
<------------------------------------Welcome to Library Manager Application--------------------------------------------->

PRESS 1 : IF YOU ARE CREATING NEW ACCOUNT  OR  PRESS 2 : IF YOU ARE AN EXISTING STUDENT
Enter Your Answer
2
<========================================EXISTING STUDENT WINDOW===================================>
PRESS 1 : TO ADD BOOK
PRESS 2 : TO BORROW BOOK
PRESS 3 : TO RETURN A BOOK
PRESS 4 : TO DISPLAY NEW BOOKS
PRESS 5 : TO MODIFY BOOK NAME
PRESS 6 : TO REMOVE A BOOK
PRESS 7 : TO DISPLAY NEW STUDENTS
PRESS 8 : TO VIEW TEAM MEMBERS NAME
8

<-----------------------------Our Team---------------------------------------->


Dharaneesh RS      CSE053
Deepan G           CSE043
Ganesh Prabu BO    CSE063



Thank You
Do you want to continue?
Press : 1 for Yes(or)Press : 0 for No
CHOOSE YOUR OPTION =>
```

SLIDE 10: CONCLUSION

In conclusion, our Library Management System is a comprehensive solution designed to streamline the management of library resources. By leveraging structures and functions

COLLABORATIVE DEVELOPMENT BY

A. DHARANEESH R S

B. DEEPAN G

C. GANESH PRABU BO