

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**PULCHOWK CAMPUS**

A Course Project Submitted to the Department of Electronics And Computer Engineering in partial fulfillment of the requirements for the practical course on Computer Programming [CT 401]

**LIBRARY MANAGEMENT SYSTEM**

|  |  |
| --- | --- |
| **SUBMITTED TO:** | **SUBMITTED BY:** |
| The Department Of Electronics and Computer Engineering,  Pulchowk Campus, Lalitpur | 1. Abiral Chalise (077BCE006) |
| 1. Anuj Kunwar (077BCE017) |
| 1. Ashma Sharma (077BCE022) |
| 1. Avishek Pokhrel (077BCE025) |

Institute of Engineering

**TABLE OF CONTENTS PAGE NO:**

Acknowledgement----------------------------------------------------------------3

Abstract-----------------------------------------------------------------------------4

Background------------------------------------------------------------------------5

Objective---------------------------------------------------------------------------6

General Theory--------------------------------------------------------------------7

Problem Analysis-----------------------------------------------------------------14

Algorithm -------------------------------------------------------------------------15

Flowchart -------------------------------------------------------------------------19

Source Code-----------------------------------------------------------------------29

Output------------------------------------------------------------------------------49

Discussion-------------------------------------------------------------------------60

Summary---------------------------------------------------------------------------62

Reference**--------------------------------------------------------------------------**63

**ACKNOWLEDGEMENT**

We have developed a library and/or book management software using C programming language that can be applicable for any type of libraries. It would have been impossible to do anything without the assistance, help and cooperation from our subject teacher, our friends and the seniors. In the process of developing the software we came across a lot of difficulties and we received proper guidance and assistance from these individuals. We would like to offer our deep thanks to all the personnel who provided us with the necessary knowledge on the preparation of this project.

We would like to thank our subject teacher, **Er. Santosh Giri** who gave us the concepts of C programming language through every lectures and provided us with necessary notes. Google has undoubtedly been our mentor throughout the completion of our project. So, it truly deserves our gratitude. The task this year has been more challenging in comparison to our seniors because of the environment and the newly born disease. Despite the fact that the group members have not even met once, we could come up to the end. For this, we would like to thank the cooperative surrounding of Pulchowk Campus. Also we would like to express heartfelt thanks to all our dear friends and our seniors for their incredible advice, encouragement and support.

We also consulted various books for the theoretical part. We would like to thank the authors of them. Some of them are listed below:

• Mr.Yashwant Kanetkar-Let Us C.

• Mr. Venugopal-Programming With C.

• Mr. Byron S. Gotterfried-Programming With C.

• Mr.Balagurusamy-Programming In ANSI C.

To sum up, we appreciate every helping hand that selflessly provided their support leading us to complete our project successfully.

**ABSTRACT**

Library management system is a project which aims in developing a computerized system to maintain the daily works of library. To keep the records of the issued books, it is very difficult to organize manually. Maintenance of all this information physically is a very complex task. Owing to the advancement of technology, organization of library becomes much simpler. The LM has been designed to computerize and automate the operations performed over the information about the students, book issues as well as returns and many other operations. This project has some features which are generally not available in normal LMS like a facility to log in. This marks the safety followed by our project. It has also a facility where after logging in, students can see list of books issued and its issue date and return date and also the students can add new books, delete it after returning, search for a specified book and many more. This computerization of library helps in many instances of its maintenance. It reduces the workload of management as most of the laboring work done is reduced.

.

Overall this project of ours is being developed to help specially the students as well as staff

of library to maintain the book records in the library in the best way possible and also to reduce the

blue collar efforts.

**Keyword**:

Blue collar

**Abbreviation:**

LM: Library Management

LMS: Library Management System

**BACKGROUND**

Being engineering students, the important factor to measure our ability and skill in any field, is through practical performances. Considering this fact, a mini-project is included in our lab assignment; it is to prepare a project using C programming language. This is a very creative way to develop our practical ability in developing programs and software. This project is a part of the subject Computer Programming –I, first year and first part of Civil Engineering course.

Our project topic is LIBRARY MANAGEMENT SYSTEM . The program stores the data related to book records like addition of books, modification in books , deletion after returning the book, etc in any library.

To build the project, it took us about two weeks. The project required vast knowledge of structure, function, array, string and data file which are the important and complex features of C programming language. We received help from our seniors, notes provided by our respected teacher, **Er. Santosh Giri** , consulted various books as well as resources available in the internet that turned out to be more than useful in collecting information about different topics used in our project .After gaining some knowledge, we started the program on modular level. With some progress on each day, the coding was completed after 10 days time. After the program was successfully written, compiled and tested for syntax and logical errors, we then prepared the documentation part. For this, we consulted our seniors again and also searched in Google about the rules and procedure of writing a report.

The group work went more interesting day by day due to the increased focus of the all of us. The work load was also easily managed because of mutual understanding and equal division of task. This kind of team work marked the completion of the project.

**OBJECTIVES**

The specificobjective of our project is:

* 1. To develop library management software based on C programming, that can perform every tasks in any library , which includes addition, modification, deletion, selection and many more records in any type of library.

The generalobjectives of our project are:

1. To be familiar with resource reusability by making user defined header files.

2. To learn about File Handling through C programming.

3. To make the program to occupy less memory as far as possible.

4. To make such a program that makes us able to work in major projects further in future.

5. To use different user defined functions, to break a program into many simplified & smaller parts

to deal with and to make it easier to understand the codes and debug.

6. To get a surface understanding of knowledge about software development.

7. To learn to work in a team and backup each other.

8. To explore the features of C language.

**GENERAL THEORY**

**Introduction to C:-**

**C** is a general purpose computer programming language developed in 1972 by Dennis Ritchie at the Bell Telephone Laboratories. C is one of the most popular programming language. It is widely implementing application software as well as system software. C is a structured language made up of collection of procedures which consists of declarations, statements and other elements. It uses English language. C resembles other high level languages like Pascal and Fortran. It gives maximum control and efficiency to the programmer so is worldwide appreciated and very common.

C programs have ability to write concise programs even though a large number of operators are used. C includes a number of extensive library functions to enhance the basic instructions. It also allows a programmer to write additional functions of their own.

**Structures of C**

C is structured programming language. It consists of:

* Pre-processor directives
* Variables and function declaration
* Main function
* Other functions

C uses its character set, reserved or key words, identifier, variables, constants, operators and punctuators.

Each function in any program must contain a function heading, a list of argument declarations and a compound statement. The compound statement are enclosed within a pair of curly braces i.e.{}. Each expression statement must end with a semi-colon. Comments or remarks can be put anywhere in a program enclosed between /\* and \*/.

**Control Statement:**

Logical operation is carried out by several symmetrical or logical statements. There are three types of control statement based on their function.

* 1. **Sequential structure:**

In sequential control statements, the program executes the instruction in the sequential order in which they are contained in the source code.

* 1. **Selective structure:**

Selective structures are used when we have a number of situations where we need to change the order of execution of statements based on certain condition. The selective statements make a decision to take the right path before changing the order of execution. C provides the following statements for selective structure:

**# if statements**

**# switch statements**

**if statements:**

The if statement is a powerful decision making statement and it is used to control the flow of execution of statements. It is a two way statement and is used in conjunction with an expression.

If statement allows the computer to evaluate the expression first and then on depending whether the value of the expression is true or false, it transfers the control to the particular statement. At this point the program has two paths to follow: one for true condition and other for false condition. The types of if statements are explained below:

**Simple if statement:**

The simple if statement is used to conditionally excite a block of code based on whether a test condition is true or false. If the condition is true the block of code is executed, otherwise it is skipped. The syntax of if statement is given below:

 if(test expression)

      {

          statement-block;

       }

         statement-x;

**if else statement:**

The if else statement extends the idea of the if statement by specifying another section of code that should be executed only if the condition is false i.e. conditional branching. True- block statements are to be executed only if the test expression is true and false block statements to be executed only if the condition is false. The syntax of if else statement is given below:

if(test expression)

 {

    true block statement;

  }

   else

  {

     false block statement;

  }

**The switch statement:**

c has built in multi way decision statement known as switch. It successively test the value of an expression against a list of case values (integer or character consonants).when a match is found the statement associated with that case is executed. The syntax of switch expression is given below:

  switch(expression)

    {

        case constant-1:

        block-1;

        break;

        case constant-2:

         block-2;

         break;

       ………….

        ………….

       case constant-n:

       block-n;

       break;

       default:

       default statement;

}

* 1. **Looping:**

Loop caused a section of code to be repeated for a specified number of times or until some condition holds true. When a condition becomes false, the loop terminates and control passes to statement below loop. Different types of loops are discussed below with their major characteristics and syntax used in C:

**for loop:**

The for loop is used to execute a block of code for a fixed number of repetitions. Initialization is generally an assignment statement used to set loop control variable. Test expression is a relational expression that determines when loop exits. Update expression defines how the loop variable changes each time when the loop is repeated. The syntax of for loop is given below:

for(initialization expression; test expression ; update expression)

  {

     body of loop;

  }

**while loop:**

The while loop specifies that a section of code should be executed while a certain condition holds true. The syntax of while loop is given below:

 while(test expression)

   {

     body of loop;

     (

statements block);

    }

**do while statement:**

the do while statement is very similar to while statement. It also specifies that a section of code should be executed while a certain condition holds true. the difference between while and do while loop is that while loop test its condition at the top of its loop but do while loop tests its condition at the bottom of loop. In while loop, if the test condition is false, the block of code is skipped. Since condition is tested at the bottom of loop in do while loop, its block of code is always executed at least once. The syntax of do while loop is given below:

do

 {

   body of loop

 }while (test expression);

**break statement:**

 The break statement is used to jump out of loop. The break statement terminates the execution of the nearest enclosing loop. Control passes to the statement that follows the terminated statement in a switch break statement causes  the program to execute the next statement after switch.

break;

**continue statement:**

continue statement is used to skip rest of the codes in the loop or blocks of statements and to move to the next iteration.

**Function:**

A function is a self contained program segment that carries out some specific well defined task. Every C program consists of one or functions. Execution of program always begins by carrying out instruction in main. Function makes program significantly easier to understand and maintain. A well written function may be reused in multiple programs. Program that are easier to design, debug and maintain.

**Return statement:**

A function may or may not send back any value to the calling function. If it does, it is through return statement. The called function can only return only one value per call at most. The syntax of return statement is given below:

Return;

**Pointer:**

 A pointer is a variable that represents the location (rather than value) of a data item, such as a variable or an array element. Pointers can be used to pass information back and forth between a function and a reference point. Pointer provides a way to return multiple data items from a function via function argument. When a pointer variable is declared, the variable name must be preceded by an aesteric (\*).the syntax of a pointer declaration is:

data type \*ptar;

**Structure:**

 It is a heterogeneous user defined data type. It is also called constructed data type. It may contain different data types .Structure can also store non homogenous data type into a single collection. Structure may contain pointet, arrays, or even other structures other than the common data types such as int, float, long int etc. A structure provides a means of grouping variables under a single name for easier handling and identification. It can be defined as new named types. It is a convenient way of grouping several pieces of related information together. Complex hierarchies can be created by nesting structures. Structures may be copied to and assigned. They are also useful in passing groups of logically related data into structures. The declaration of structures is given below:

 struct tag

{

   member 1;

    member 2;

     member n;

};

**File:**

Many applications require that information be written to or read from an auxiliary memory device. Such information is stored on the memory device in the form of a data file. The data files allow us to store information permanently and to access and alter that information whenever necessary.

**Opening a file:**

  Before performing any input / output operation, file must be opened. While opening file, the following must be specified:-

1. The name of file.
2. The manner in which it should be opened (that for reading ,writing ,both reading and writing ,appending at the end of file, overwriting  the file)
3. When working with a stream oriented data file ,the first step is to establish a buffer area, where information is temporary stored while being transferred between the computer’s memory and data file .the buffer area is established by writing

**FILE \*ptvar;**

where File is a special structure type  establishes the buffer area and ptvar is a pointer variable that indicates the beginning of the buffer area the library function fopen is used to open a file .This function is used to open a file .This function is typically written as

            ptvar=fopen(file name, file type );

where file name and file type are strings that represent the name of the data file and the manner in which the data file will be utilized.

Finally, a file can be closed at the end of the program. This can be accomplished with the library function fclose. The syntax is simply,

   fclose(ptvar);

    Processing a file:

    Most data file application requires that a data file be altered as it is being processed. For example, in an application involving the processing of customer records, it may be desirable to add new records to the file, to delete the existing records, to modify the contents or to rearrange the records.

**File type                                                                                                    Meaning**

**“r”** open an existing file for reading only.

**“w”** open a new file for writing only. If the file with specified file\_name currently exists it will be destroyed and new file is created in its place.

**“a”** open an existing file for appending. A new file will be created if the file with the specific file\_name does not exist.

**“r+”** open an existing file for both reading and writing.

**“w+”** open a new file for both reading and writing. If a file with the specified file\_name currently exists, it will be destroyed and a new file is created in its place.

**“a+”** open an existing file for reading and writing. A new file will be created if the file with the specified filename does not exist.

**The fread and fwrite functions :**

Some applications involve the use of data files to store blocks of data, where each block consists of a fixed number of contiguous bytes. Each block will generally represent a complete data structure, such as a structure or an array. for such applications it may be desirable to read the entire block from the data file or write the entire block to the file.

The library function fread and fwrite are intended to be used in situations of this size of the data block, the number of data block being transferred and file pointer. thus typical fwrite and fread functions :

**Fwrite(&customer, sizeof(record),1,fpt);**

**Fread(&customer, sizeof(record),1,fpt)**

Where customer is a structure variable of type record and fpt is the file pointer associated with the data file that has been opened for input/output. Once an unformatted data file has been created with fwrite, it can be read with fread function. The function returns a zero value if an end-of-file condition has been detected and non-zero value if an end-of-file is not detected. Hence, a program that reads an unformatted data file can be reading file, as long as the value returned by fread is non-zero value.

**rename:**

The **rename function** changes the name of a file. You must close the file before renaming, as a file cannot be renamed if it is open.

The syntax for the rename function is:

int rename(const char \*old, const char \*new);

**old :**The old file name that will be changed.

**new:** The new file name to use.

**fflush():**The function fflush(stdin) is used to flush the output buffer of the stream. It returns zero, if successful otherwise, returns EOF and feof error indicator is set.

The syntax of fflush(stdin) is:

int fflush(FILE \*stream);

**remove():**

In the C Programming Language, the **remove function** removes a file pointed to by filename.

**Syntax**

The syntax for the remove function in the C Language is:

int remove(const char \*filename);

**filename:** The file to delete.

The remove function returns zero is successful, otherwise nonzero.

**PROBLEM ANALYSIS**

**1. Understanding the problem**

All the members of the group held the meeting and decided to make a program for **Library Records Management System**. The first problem that raised at the decision to make Library Record Management program was how to make a login page, search for a particular student details and how to update them.

**2. Input Requirements**

This section consists of the source of input data, which should yield the meaningful output. The source code is the major input requirement for the Library Record Management program. The code should be clear so that the student/librarian can easily understand what the input should be. The program starts by displaying a login screen which asks user to enter the username and password. After successful login, the program displays the list of options where the user has to press 1,2, ... depending upon what the user want to do.

**3. Output Requirements**

Before programming we must know what should be the output of the program. In this Library Record Management program, the output contains displaying the record of the user on the basis of choice of the user.

**4. Processing Requirements**

The processing requirement should be clearly defined to convert the given input data to the required output. In this project 4 people collaborated together for the completion for the project. The coding is done on VS code.

**5. Technical Feasibility**

In the context of feasibility the coding didn’t require that much hard work and manpower due to simplicity of the program. Though the program was short, it wasuser interactive and it yield the required output.

**ALGORITHM**

To write a logical step-by-step method to solve the problem is called algorithm, in other words, an

Algorithm is a procedure for solving problems. In order to solve a mathematical or computer

problem, this is the first step of the procedure. An algorithm includes calculations, reasoning and

data processing. Algorithms can be presented by natural languages, pseudo code and flowcharts,

etc.

**Algorithm of the program:**

1. Start
2. Declare file pointers \*f (for storing information of books) and \*lp (for storing login credentials).
3. Declare character filename.
4. Declare structures:
5. library1 (for storing book information) containing character book\_title ,author\_name, integer acc , character category, sname, sroll and integers day, month, year.Then initialize ir as library.
6. login (for storing login information) containing characters username and password. Then initialize it as l.
7. Declare function prototypes:
8. Headmessage (to aware user what tab they are using)
9. Welcome message (to display welcome screen beautifully)
10. init (for loading login credentials)
11. login (for logging in process)
12. MainMenu (for main part of program for managing library management)
13. Update (for updating login credentials)
14. Add book (for adding book information in file)
15. Disp\_book (for displaying book information from file)
16. Delete book (for deleting book information from file)
17. Search (for searching book information from file)
18. Modify (for modifying book information in file)
19. Proceed In the main function.
20. Call welcomemessage function and display welcome message
21. Call headmessage function and display headmessage
22. Return to step 7 and proceed from 7.d.
23. Call init function(control of program goes to init function)
24. Proceed in init function
25. Call headmessage function and display headmessage
26. Return to step 8 and proceed from 8.c
27. Declare file pointer \*lp.
28. Check if “login.txt” file exists.

🡺if yes, then assign the memory location of file to pointer lp.

Open the file in read mode and read the login credentials.

Close the file.

🡺if no, then copy user as username an pass as password for login credentials.

1. Return to main function
2. Call login function(control of program goes to login function)
3. Proceed to login function
4. Call headmessage function and display headmessage
5. Return to step 10 and proceed from 10.c.
6. Declare characters:user and pass.
7. Initialize l=0.
8. Open file “login.txt” in read mode and assign the memory location of file to pointer lp and read tha data of structure login(read login credentials).
9. Display “login screen”.
10. Read username and password form user.
11. If the login credentials are not entered correctly

🡺Allow total three times to enter correct login credentials. For at most three wrong input credentials display “login failed” and terminate the program

1. If the login credentials are entered correctly

🡺Display “Logged into the system ”

🡺Call mainmenu function (control of program goes to mainmenu function)

1. Proceed to mainmenu function
   1. Call headmessage function and display headmessage
2. Return to step 11 and proceed from 11.c.
3. Declare integer variable choice.
4. Copy libraryrecords as the name for file “filename”.
5. Display:

Mainmenu!!!!

1. Add Book Information

2. Display Book Information

3. Delete Book Records

4. Search Specified Book

5. Modify Records

6. Update login information

0. Exit

1. Use Switch in which  case values are used as argument and above cases may occur  according the “choice” variable.
2. If all above case values doesnot match choice value;a default  statement is set which displays “Enter 0,1,2,3,4,5,6”and if default is encountered then goto main menu.
3. If **choice🡪1**; **add\_book** function is called and proceeded as;
4. Call headmessage function and display headmessage
5. Return to step 12 and proceed from 12.c.
6. Declare  variables test as  char  type     .
7. Open the file“libraryrecords” in append and write mode
8. If file cannot open then exit
9. Else enter the book records
10. Get the book title ,author name,acc no, category of book, Student’s name(one who has taken the book),student”s roll no and date of issue.
11. Write to the file.
12. If want to stop adding books and goto main menu, then press esc key. Else, goto step 12. D .
13. Close that file.
14. Return to mainmenu
15. If **choice🡪2**; **disp\_book** function is called and proceeded as;
16. Call headmessage function and display headmessage
17. Return to step 13 and proceed from 13.c.
18. Open the file “libraryrecords” in read mode.
19. If the file doesnot exist  print file cannot be opened
20. Read all the file record
21. Display all book  records
22. Close that file
23. Return to mainmenu
24. If**choice🡪3**; **delete\_book** function is called and proceeded as;
25. Call headmessage function and display headmessage
26. Return to step 14 and proceed from 14.c.
27. Declare file pointer \*temp and assign its name to “tempfile”
28. Enter the acc no of book to be deleted
29. Open the file “libraryrecords” in read mode
30. If file can’t  open ,print cannot open the file and exit
31. Open new temporary file(\*temp) in write mode
32. If cannot open the file, then exit
33. If “libraryrecords” opens ,then, read file
34. Check for acc no of the book and entered acc no

-If true go to 14.j

-Else write the datas of “libraryrecords” to new file ”tempfile”

1. Close both files
2. Remove file ““libraryrecords”
3. Rename “tempfile” as “libraryrecords”
4. Return to mainmenu
5. If **choice🡪4**; **search** function is called and proceeded as;
6. Call headmessage function and display headmessage
7. Return to step 15 and proceed from 15.c.
8. Open the file in r mode
9. If file cannot open, then, exit
10. Here we can search the book record information in different modes on the basis of name of name of book(N/n) OR students roll no(R/r) OR accession code of book(C/c)
11. Use switch statement for choosing any of searching mode
12. For the particular information entered in searching mode;
    * 1. 🡺if such record about book is present then, read the library structure file and print the record of that book
      2. 🡺if such record is not there then display “no such records”
13. Close the file.
14. Return to mainmenu
15. If **choice🡪5**; **modify** function is called and proceeded as;
16. Call headmessage function and display headmessage
17. Return to step 16 and proceed from 16.c.
18. Declare file “tempfile”
19. Declare integer a
20. Enter the accession code of the book to be modified
21. Open the file “libraryrecords” in r mode
    * 1. 🡺if  file cannot open then, exit
22. Open the file “tempfile” in w mode
    * 1. 🡺if  file cannot open then, exit
23. Read the accession code of book to be modified
24. Read the data from the file ”libraryrecords” using fread
25. Compare the accession code from file “libraryrecords” and accession code entered by user
    * 1. 🡺if true, then modify the records of the book and write this new record in the file “tempfile” and goto 16.j
      2. 🡺if not true, then copy old records from file “libraryrecords” to “tempfile”
26. Close both files
27. Remove “libraryrecords”
28. Rename “tempfile” as “libraryrecords"
29. Return to mainmenu
30. If **choice🡪6**; **update** function is called and proceeded as;
31. Call headmessage function and display headmessage for update function
32. Return to step 17 and proceed from 17.c.
33. Open “login” file in “w” mode
34. Read new username and password from user
35. Write new username and password to the file “login” using fwrite
36. Close the file “login”
37. Inform user that username and password has been changed
38. End the program and make user use updated login credentials for further login
39. If **choice🡪0**; **exit** function is called and program is terminated.
40. stop

**FLOWCHART**

A flowchart is the graphical or pictorial representation of an algorithm with the help of different

symbols, shapes and arrows in order to demonstrate a process or a program. With algorithms, we

can easily understand a program. The main purpose of a flowchart is to analyze different processes.

**Several standard graphics that are applied in a flowchart:**

**▪ Terminal Box - Start / End:**

**▪ Input / Output:**

**▪ Process / Instruction:**

**▪ Decision:**

**▪ Arrow:**

**▪ Connector:**

**▪ Function call:**

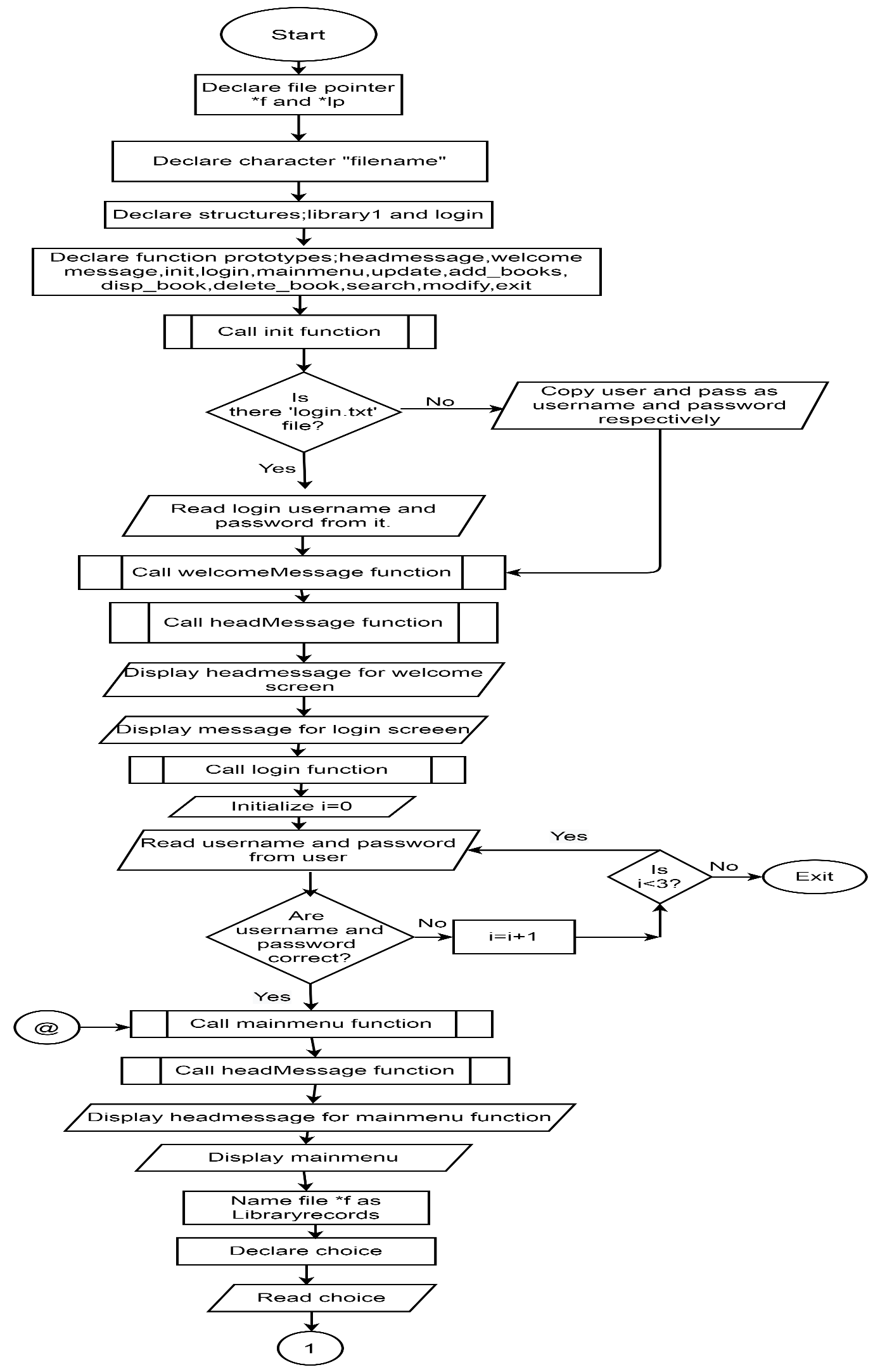
The graphics above represent different part of a flowchart. The process in a flowchart can be

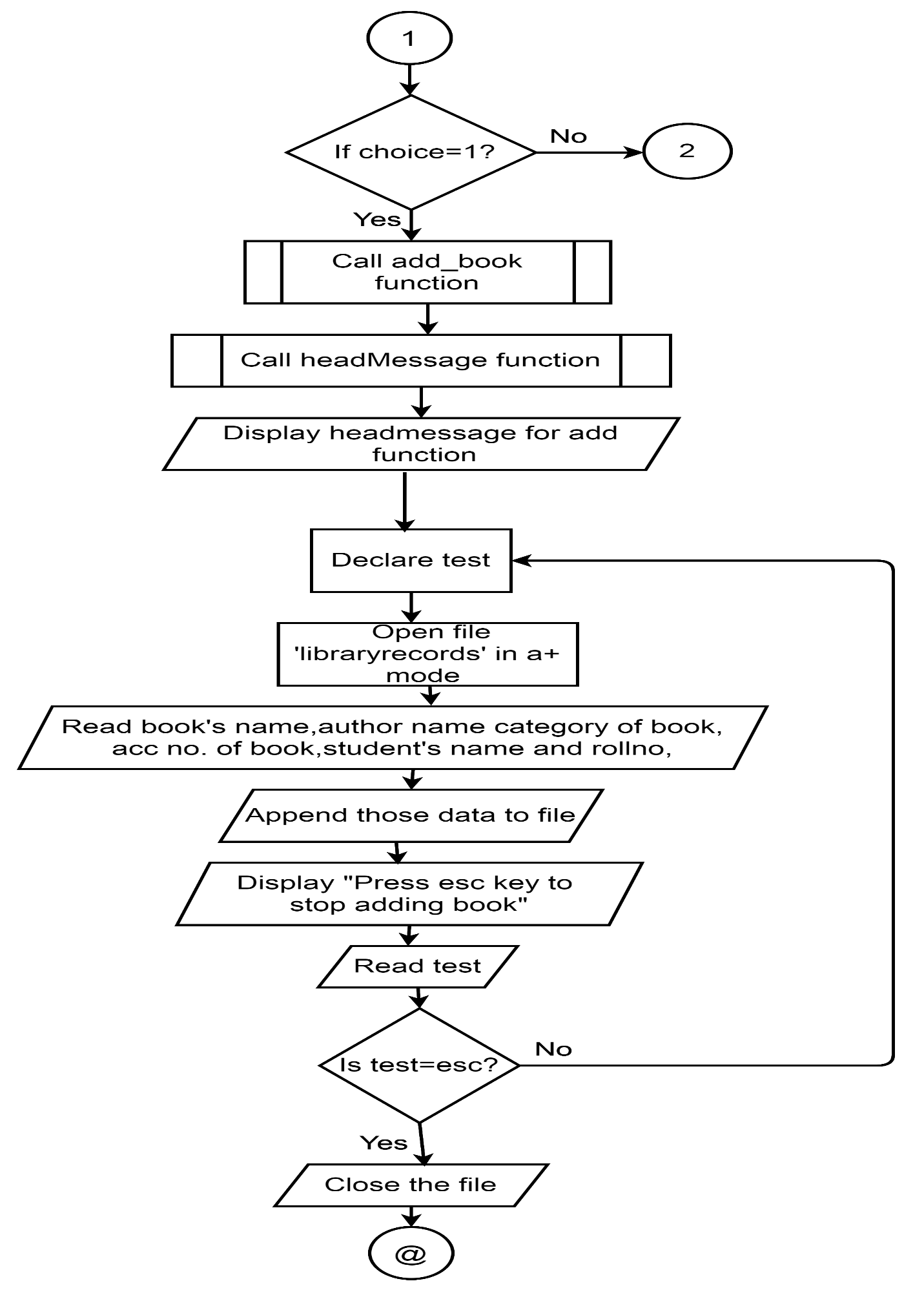
expressed through boxes and arrows with different sizes and colors. In a flowchart, we can easily

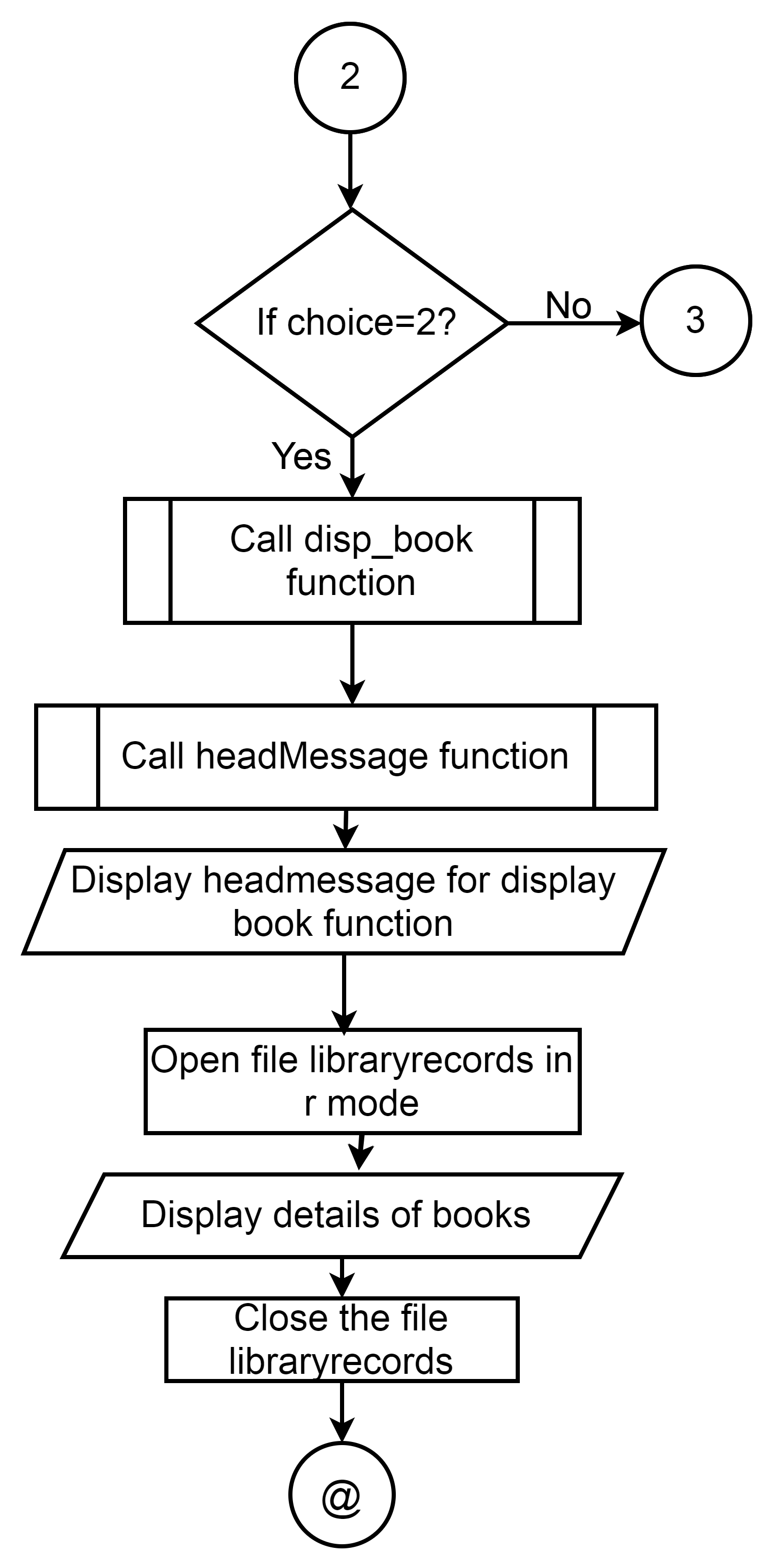
highlight a certain element and the relationships between each part.

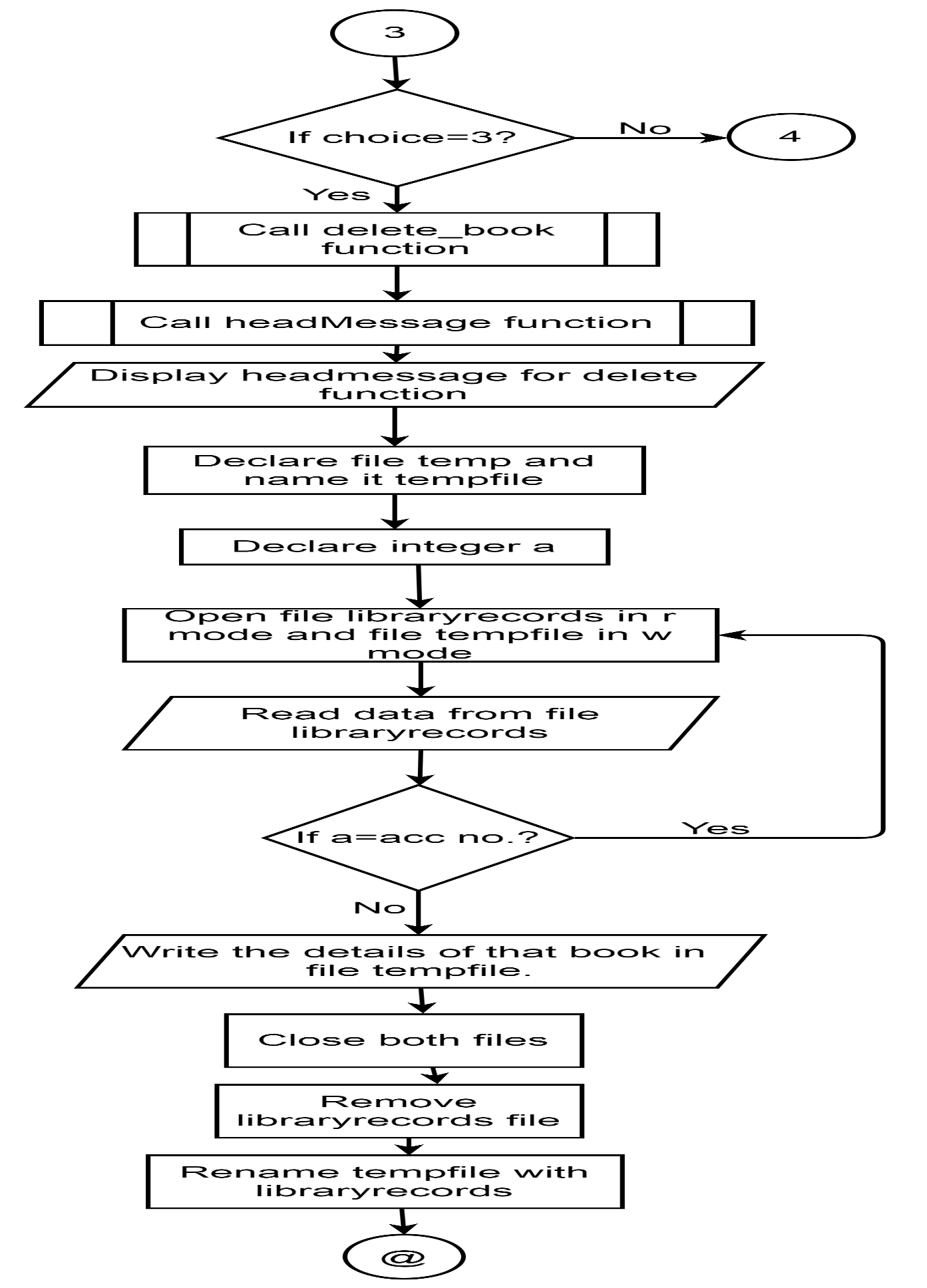
**Flowchart of the program:**

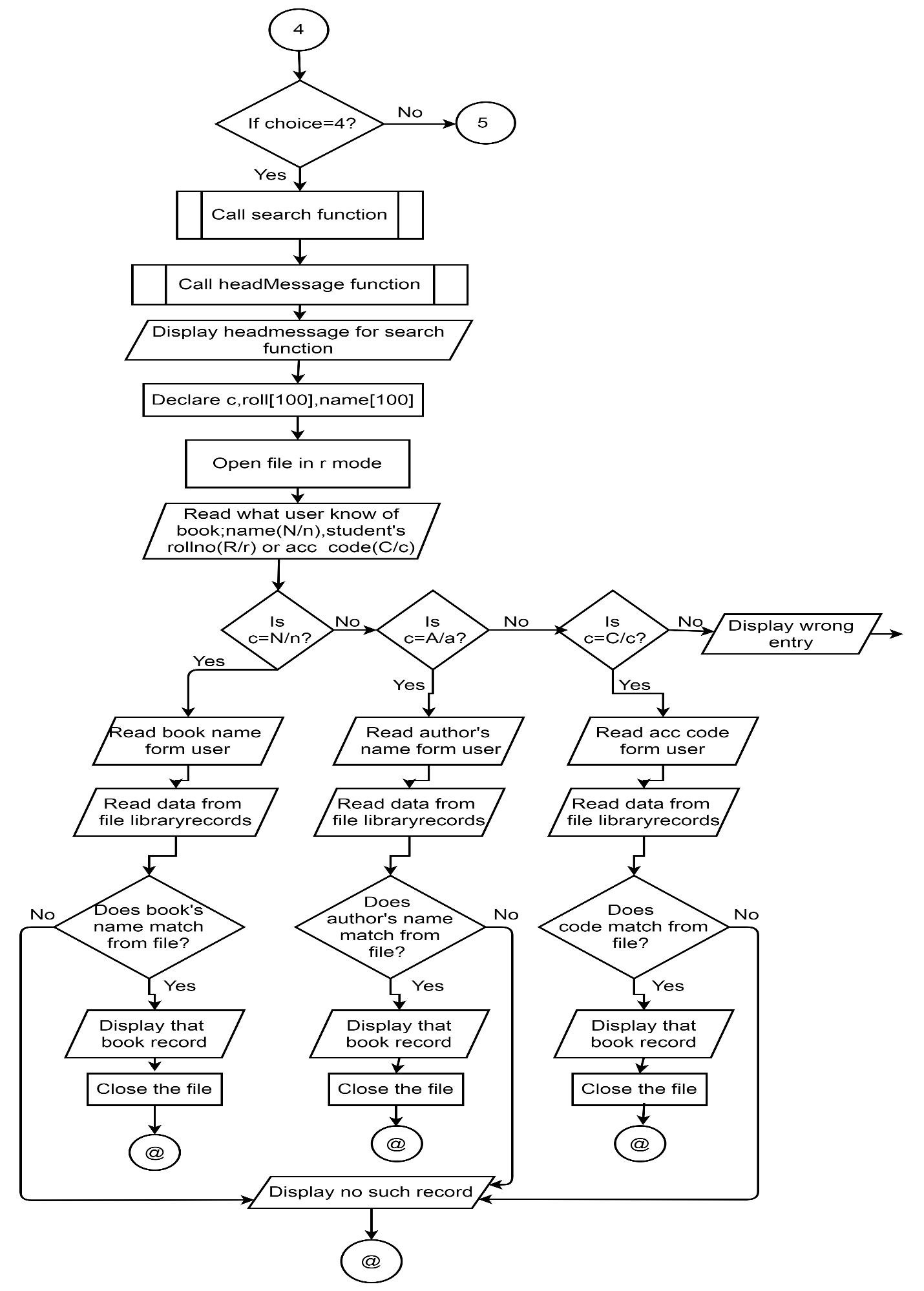
The block diagram of the program that we have developed is drawn in the following pages.

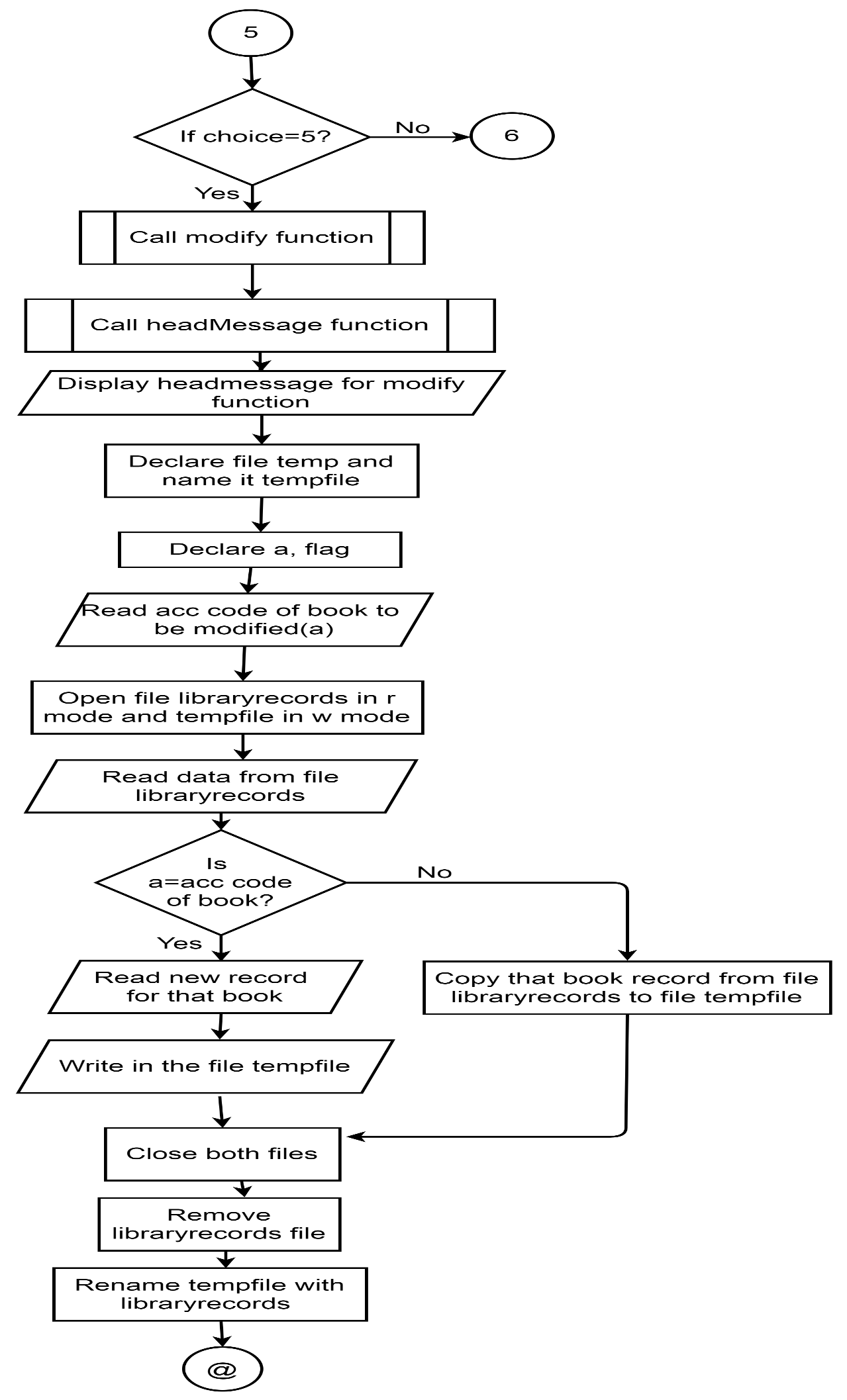


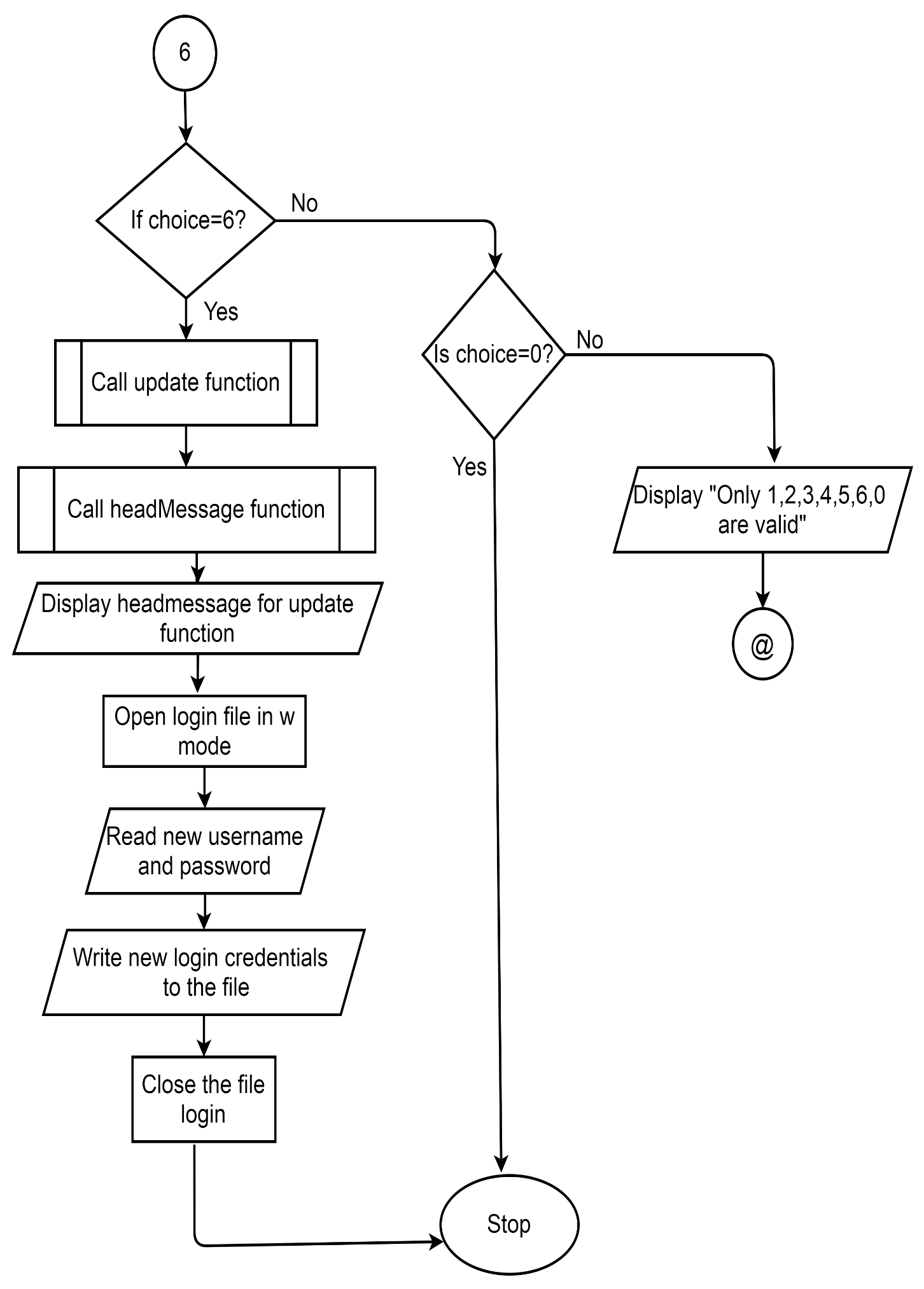












**SOURCE CODE**

//documentation section//

/\* Library management system

Developers:

1. Abiral Chalise (077BCE006)

2. Anuj Kunwar (077BCE017)

3. Ashma Sharma (077BCE022)

4. Avishek Pokhrel (077BCE025)

Date: 2021/8/7 \*/

//link section//

#include<stdio.h>//for standard functions

#include<string.h>//for string handling functions

#include<stdlib.h>//for using :exit(); and system("cls"); functions

#include<conio.h>//for using :getch(); function

//definition section//

#define clrscr(); system("cls"); //Is to be done in VS code(modern C complier) in older compliers it is pre defined in header file #include<conio.h>.

//global declaration section//

FILE \*f;//for storing information of books

FILE \*lp;//for storing login information

struct library1//for storing book information

{

char book\_title[100];

char author\_name[100];

int acc ;

char category[20];

char sname[50];

char sroll[50];

int day,month,year;

}library;

struct login{//for storing login information

char username[50];

char password[50];

}l;

char filename[30];//for storing file name

//function prototyping//

void headMessage(char \*);

void welcomeMessage();

void init();//loading login credentials

void login();//log in process

void mainmenu();//main part of program for managing library management

void update();//update login credentials

void add\_book();//adding book information in file

void disp\_book();//displaying book information from file

void delete\_book();//deleting book information from file

void search();//searching book information from file

void modify();//modifing book information in file

// main() function section//

int main()

{

init();

welcomeMessage();

login();

return 0;

}

//sub program section//

void headMessage(char \*message)

{

clrscr();

printf("\t\t\t###########################################################################");

printf("\n\t\t\t############ ############");

printf("\n\t\t\t############ Library management System Project in C ############");

printf("\n\t\t\t############ ############");

printf("\n\t\t\t###########################################################################");

printf("\n\t\t\t---------------------------------------------------------------------------\n");

printf("\t\t\t\t\t\t%s",message);

printf("\n\t\t\t----------------------------------------------------------------------------\n");

}

void welcomeMessage()

{

headMessage("www.pulchowklibrarysystem.com");

printf("\n\n\n\n\n");

printf("\n\t\t\t \*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*\n");

printf("\n\t\t\t =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=");

printf("\n\t\t\t = WELCOME =");

printf("\n\t\t\t = TO =");

printf("\n\t\t\t = LIBRARY =");

printf("\n\t\t\t = MANAGEMENT =");

printf("\n\t\t\t = SYSTEM =");

printf("\n\t\t\t =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=");

printf("\n\t\t\t \*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*-\*\*\n");

printf("\n\n\n\t\t\t Enter any key to continue.....");

getch();

}

void init()

{

if( lp=fopen("login.txt","r")){//look if the file exist

fread(&l,sizeof(l),1,lp);//read the credentails from file if exist

fclose(lp);

}

else{

strcpy(l.username,"user");//assign default crededtials if file doesn't exist

strcpy(l.password,"pass");

}

}

void login()

{

int L=0;

char user[50], pass[50];

headMessage("Login Screen");

do

{

printf("\n\t\t\t\tEnter Username and Password");

printf("\n\t\t\t\tUsername:");

scanf(" %[^\n]" ,user);

printf("\n\t\t\t\tPassword:");

scanf(" %[^\n]" ,pass);

if((strcmp(user,l.username)==0)&& (strcmp(pass,l.password)==0))//check if the username and password are correct

{

printf("\n\t\t\t\tLogged Into System.");

printf("\n\t\t\t\tPress any key to go next.....");

printf("\n------------------------------------------------------------------------------------------------");

getch();

mainmenu();//calling mainmenu function if the credentials are correct

}

else

{

printf("\t\t\t\tLogin Failed Enter Again Username & Password Again\n\n");

L++;

}

}while(L<3);

if(L>=3){//wrong credentials multiple times

headMessage("Login Failed");

printf("\t\t\tSorry,Unknown User.");

getch();

}

}

void mainmenu()

{

int choice;

strcpy(filename,"libraryrecords.txt");//giving filename

do

{

headMessage("MAIN MENU");

printf("\n\n\t\t\t1.Add Book Information");

printf("\n\n\t\t\t2.Display Book Information");

printf("\n\n\t\t\t3.Delete Book Records");

printf("\n\n\t\t\t4.Search Specified Book");

printf("\n\n\t\t\t5.Modify Book Records");

printf("\n\n\t\t\t6.Update Login Information");

printf("\n\n\t\t\t0.Exit");

printf("\n\n\n\t\t\t Enter choice => ");

scanf("%d",&choice);

switch(choice)

{

case 1:

add\_book();

break;

case 2:

disp\_book();

break;

case 3:

delete\_book();

break;

case 4:

search();

break;

case 5:

modify();

break;

case 6:

update();

break;

case 0:

printf("\n\n\n\t\t\t\tThank you!!!\n\n\n\n\n");

getch();

exit(0);

default:

headMessage("MAIN MENU");

printf("\t\t\tTry valid character: 0 1 2 3 4 5 6");

printf("\n\n\n\t\t\tpress any key to continue");

getch();

}

} while(choice!=0);

}

void add\_book()

{

char test;

if((f=fopen(filename,"a+"))==NULL)

{

printf("Cannot open the file");

exit(1);

}

while(1)

{

headMessage("ADD NEW BOOKS");

printf("\n\n\n\n\n\t\t\tEnter the details of the book:\n");

fflush(stdin);//clear (or flush) the output buffer.

printf("\n\n\n\t\t\tName:");

scanf(" %[^\n]",library.book\_title);

printf("\n\t\t\tAuthor:");

scanf(" %[^\n]",library.author\_name);

printf("\n\t\t\tAccession no:");

scanf(" %d",&library.acc );

printf("\n\t\t\tCategory:");

scanf(" %[^\n]",library.category);

printf("\n\t\t\tStudent's name:");

scanf(" %[^\n]",library.sname);

printf("\n\t\t\tStudent's roll:");

scanf(" %[^\n]",library.sroll);

printf("\n\t\t\tDate of issue(day-month-year):");

scanf(" %d-%d-%d",&library.day,&library.month,&library.year);

fwrite(&library,sizeof(library),1,f);

fflush(stdin);

printf("\n\n\n\t\t\t\tEnter esc key to exit");

test=getch();

if(test==27)//The ASCII value of escape key (ESC) is 27.

break;

}

fclose(f);

}

void disp\_book()

{

if((f=fopen(filename,"r"))==NULL)

{

printf("Cannot open the file");

exit(1);

}

while((fread(&library,sizeof(library),1,f))==1)

{

headMessage("DISPLAY BOOKS DETAILS");

printf("\n\n\t\t\tDETAILS OF BOOKS IN LIBRARY\n");

printf("\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\n\n\t\t\tName:%s\n\n\n\t\t\tAuthor:%s\n\n\n\t\t\tAccesion No:%d\n\n\n\t\t\tCategory:%s\n\n\n\t\t\tStudent's name:%s\n\n\n\t\t\tStudent's roll:%s\n\n\n\t\t\tDate of issue(day-month-year):(%d-%d-%d)"

,library.book\_title,library.author\_name,library.acc ,library.category,library.sname,library.sroll,library.day,library.month,library.year);

getch();//Hold the screen until user press any key.

}

headMessage("DISPLAY BOOKS DETAILS");

printf("\n\n\t\t\tDETAILS OF BOOKS IN LIBRARY\n");

printf("\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\n\n\n\n\n\t\t\t::No Records Available::");

printf("\n\n\n\t\t\tpress any key to continue");

getch();

fclose(f);

}

void delete\_book()

{

FILE \*temp;//for storing all inforamtions except the the inforamtion to be deleted

int flag;

int a;//for storing given accession code

headMessage("Delete Books Details");

printf("\n\n\n\t\t\tEnter the accession no of the book to be deleted:\n\n\t\t\t\t");

fflush(stdin);

scanf(" %d",&a);

f=fopen(filename,"r");

if(f==NULL)

{

printf("Cannot open the file");

exit(1);

}

temp=fopen("tempfile","w");

if(temp==NULL)

{

printf("Cannot open the file");

exit(1);

}

flag=1;//initialized to 1

while(fread(&library,sizeof(library),1,f)==1)

{

if(a==library.acc)

{

flag=0;//will be 0 if the given accesion no. match

continue;//writing is skipped for given accession no's book informations

}

else

fwrite(&library,sizeof(library),1,temp);//temp contains all the informations except the one should be deleted

}

fclose(f);

fclose(temp);

remove(filename);//deleting the file initailly containing all the inforamtion including the one to be deleted

rename("tempfile",filename);//the updated file after deleting renamed as the initail file i.e updated file replacing the inital file

if(flag==0)

{

printf("\n\n\n\t\t\tRecord Deleted");

}

else

{

printf("\n\n\n\t\t\t::No such Record::");

}

printf("\n\n\n\t\t\tpress any key to continue");

getch();

}

void search()

{

char c,name[100],roll[100];

int flag,a;//a for storing given accession code

f=fopen(filename,"r");

if(f==NULL)

{

printf("Cannot open the file");

exit(1);

}

headMessage("SEARCH BOOKS");

printf("\n\n\n\n\t\t\tWhat do you know about the book?\n\n\n\n\t\t\t1. The name of the book(N/n)\n\n");

printf("\t\t\t2. The roll of the student(R/r)\n\n\t\t\t3. The accesion code of the book(C/c)\n\t\t\t");

printf("\n\n\n\t\t\t Enter choice => ");

scanf(" %c",&c);

switch(c)

{

case 'N':

case 'n':

headMessage("SEARCH BOOKS");

printf("\n\n\n\n\t\t\tEnter the name of the book:");

fflush(stdin);

scanf("%[^\n]",name);

flag=1;//flag intialized to 1

while(fread(&library,sizeof(library),1,f)==1)

{

if((strcmp(library.book\_title,name))==0)

flag=0 ;//flag will be 0 if match for searched inforamtion exist

else

continue;

if(flag==0)//displaying the searched inforamtion

{

printf("\n\n\n\t\t\tName:%s\n\n\n\t\t\tAuthor:%s\n\n\n\t\t\tAccesion No:%d\n\n\n\t\t\tCategory:%s\n\n\n\t\t\tStudent's name:%s\n\n\n\t\t\tStudent's roll:%s\n\n\n\t\t\tDate of issue(day-month-year):(%d-%d-%d)"

,library.book\_title,library.author\_name,library.acc ,library.category,library.sname,library.sroll,library.day,library.month,library.year);

printf("\n\n\n\t\t\tpress any key to continue");

getch();

headMessage("SEARCH BOOKS");

}

}

if(flag==1)

{

printf("\n\n\n\t\t\t::No such Record::");

printf("\n\n\n\t\t\tpress any key to continue");

getch();

}

break;

case 'R':

case 'r':

headMessage("SEARCH BOOKS");

printf("\n\n\n\n\t\t\tRoll of the student:");

fflush(stdin);

scanf("%[^\n]",roll);

flag=1;

while(fread(&library,sizeof(library),1,f)==1)

{

if((strcmp(library.sroll,roll))==0)

flag=0 ;

else

continue;

if(flag==0)

{

printf("\n\n\n\t\t\tName:%s\n\n\n\t\t\tAuthor:%s\n\n\n\t\t\tAccesion No:%d\n\n\n\t\t\tCategory:%s\n\n\n\t\t\tStudent's name:%s\n\n\n\t\t\tStudent's roll:%s\n\n\n\t\t\tDate of issue(day-month-year):(%d-%d-%d)"

,library.book\_title,library.author\_name,library.acc ,library.category,library.sname,library.sroll,library.day,library.month,library.year);

printf("\n\n\n\t\t\tpress any key to continue");

getch();

headMessage("SEARCH BOOKS");

}

}

if(flag==1)

{

printf("\n\n\n\t\t\t::No such Record::");

printf("\n\n\n\t\t\tpress any key to continue");

getch();

}

break;

case 'C':

case 'c':

headMessage("SEARCH BOOKS");

printf("\n\n\n\n\t\t\tAccesion No:");

fflush(stdin);

scanf("%d",&a);

flag=1;

while(fread(&library,sizeof(library),1,f)==1)

{

if(library.acc==a)

flag=0 ;

else

continue;

if(flag==0)

{

printf("\n\n\n\t\t\tName:%s\n\n\n\t\t\tAuthor:%s\n\n\n\t\t\tAccesion No:%d\n\n\n\t\t\tCategory:%s\n\n\n\t\t\tStudent's name:%s\n\n\n\t\t\tStudent's roll:%s\n\n\n\t\t\tDate of issue(day-month-year):(%d-%d-%d)"

,library.book\_title,library.author\_name,library.acc ,library.category,library.sname,library.sroll,library.day,library.month,library.year);

printf("\n\n\n\t\t\tpress any key to continue");

getch();

headMessage("SEARCH BOOKS");

}

}

if(flag==1)

{

printf("\n\n\n\t\t\t::No such Record::");

printf("\n\n\n\t\t\tpress any key to continue");

getch();

}

break;

default:

headMessage("SEARCH BOOKS");

printf("\n\n\t\t\tWRONG ENTRY");

printf("\n\n\t\t\tTRY ANY VALID CHARACTER:(N,R,C)\n");

getch();

}

fclose(f);

}

void modify()

{

FILE \*temp;

int flag;

int a;

headMessage("MODIFY BOOKS DETAILS");

printf("\n\n\t\t\tEnter the accession no of the book to be modified:\n\n\t\t\t\t");

fflush(stdin);

scanf(" %d",&a);

f=fopen(filename,"r");

if(f==NULL)

{

printf("Cannot open the file");

exit(1);

}

temp=fopen("tempfile","w");

if(temp==NULL)

{

printf("Cannot open the file");

exit(1);

}

flag=1;

while(fread(&library,sizeof(library),1,f)==1)

{

if(a==library.acc)

{

flag=0;

continue;

}

else

fwrite(&library,sizeof(library),1,temp);

}

if(flag==0){

printf("\n\n\t\t\t\t::Enter new records::");

printf("\n\n\t\t\t\t~~~~~~~~~~~~~~");

printf("\n\n\n\n\t\t\tBook Name: ");

fflush(stdin);

scanf("%[^\n]",library.book\_title);

printf("\n\t\t\tAuthor Name:");

fflush(stdin);

scanf("%[^\n]",library.author\_name);

printf("\n\t\t\tAccession No:",library.acc);

fflush(stdin);

scanf("%d",&library.acc);

printf("\n\t\t\tCategory:",library.category);

fflush(stdin);

scanf("%[^\n]",library.category);

printf("\n\t\t\tStudent's name:");

fflush(stdin);

scanf("%[^\n]",library.sname);

printf("\n\t\t\tStudent's roll:");

fflush(stdin);

scanf("%[^\n]",library.sroll);

printf("\n\t\t\tDate of issue(day-month-year)");

fflush(stdin);

scanf(" %d-%d-%d",&library.day,&library.month,&library.year);

fflush(stdin);

fwrite(&library,sizeof(library),1,temp);

printf("\n\n\n\t\t\tRecord modified");

}

else

printf("\n\n\n\t\t\t::No such Record::");

fclose(f);

fclose(temp);

remove(filename);

rename("tempfile",filename);

printf("\n\n\n\t\t\tpress any key to continue");

getch();

}

void update()

{

lp=fopen("login.txt","w");

if(lp==NULL)

{

printf("Cannot open the file");

exit(1);

}

headMessage("Update Credential");

printf("\t\t\tEnter new username: \t");

scanf(" %[^\n]",l.username);

printf("\t\t\tEnter new password: \t");

scanf(" %[^\n]",l.password);

fwrite(&l,sizeof(l),1,lp);

fclose(lp);

fflush(stdin);

printf("\n\t\t\tUsername and Password has been changed successfully");

printf("\n\t\t\tLogin Again:");

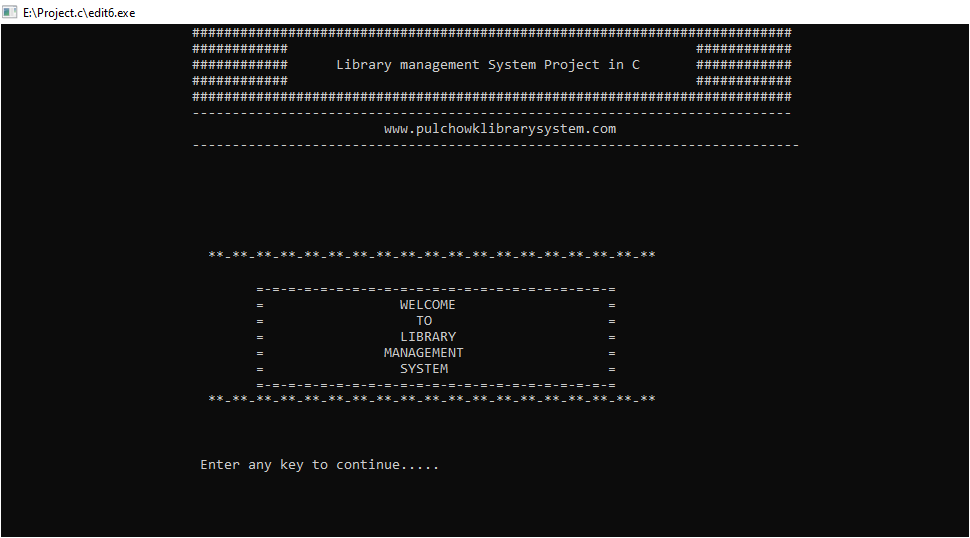
getchar();

exit(1);

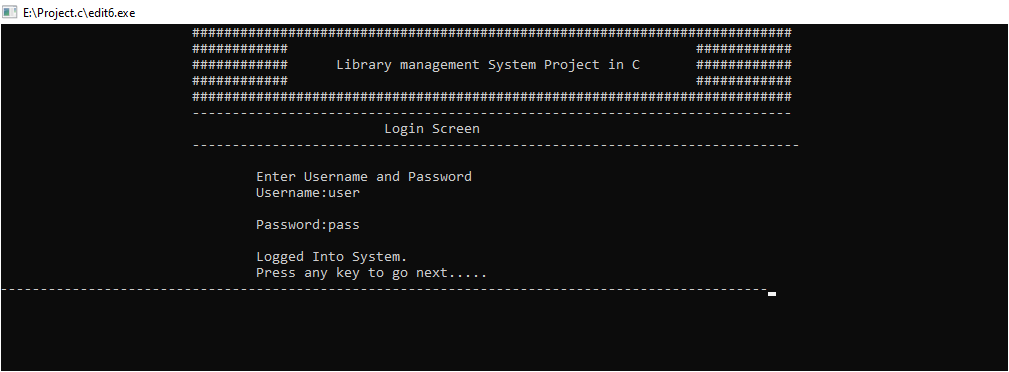
}

**OUTPUT**

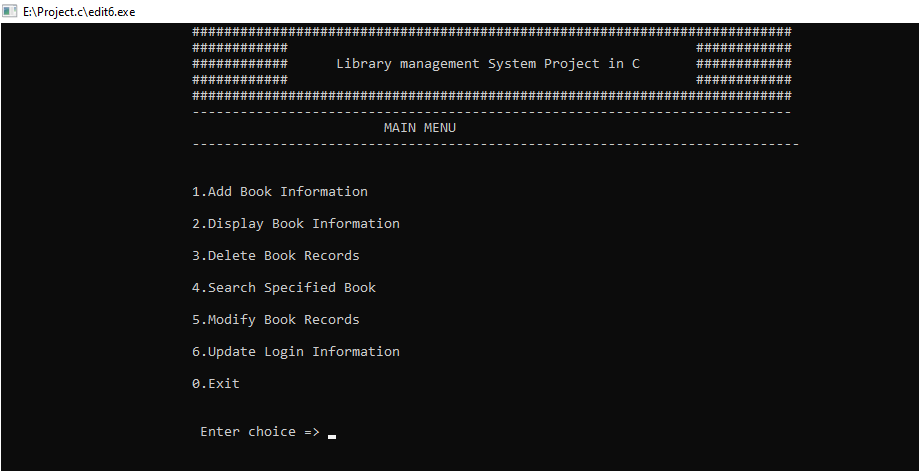
Some screenshots are attached herewith to under how the program works:

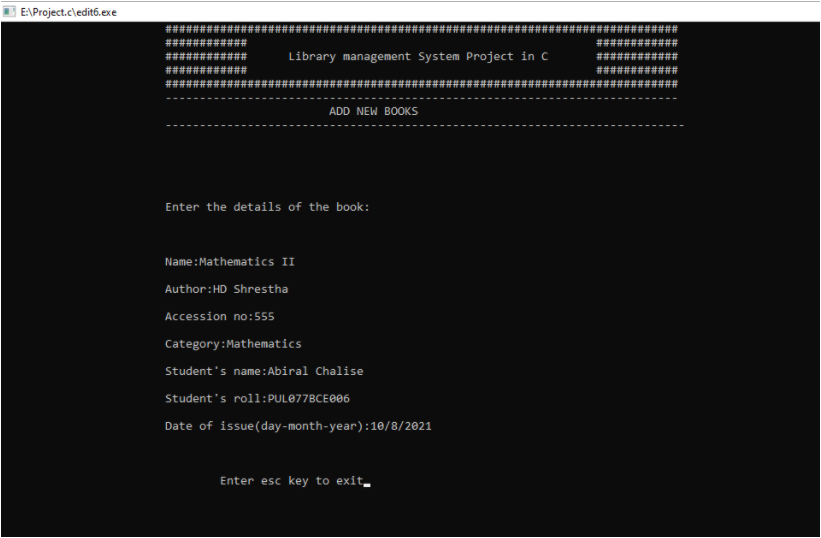


This is the welcome screen of our system. On entering any key, we get to the login screen.

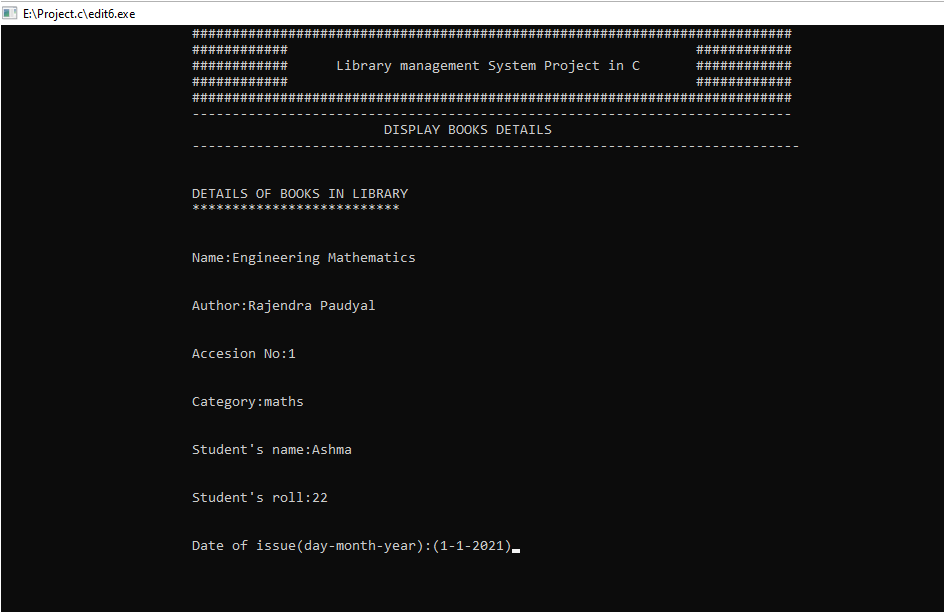


Here, we can see user can log into the system only when s/he presses the correct username and password which are user and pass initially. It can be changed later on.

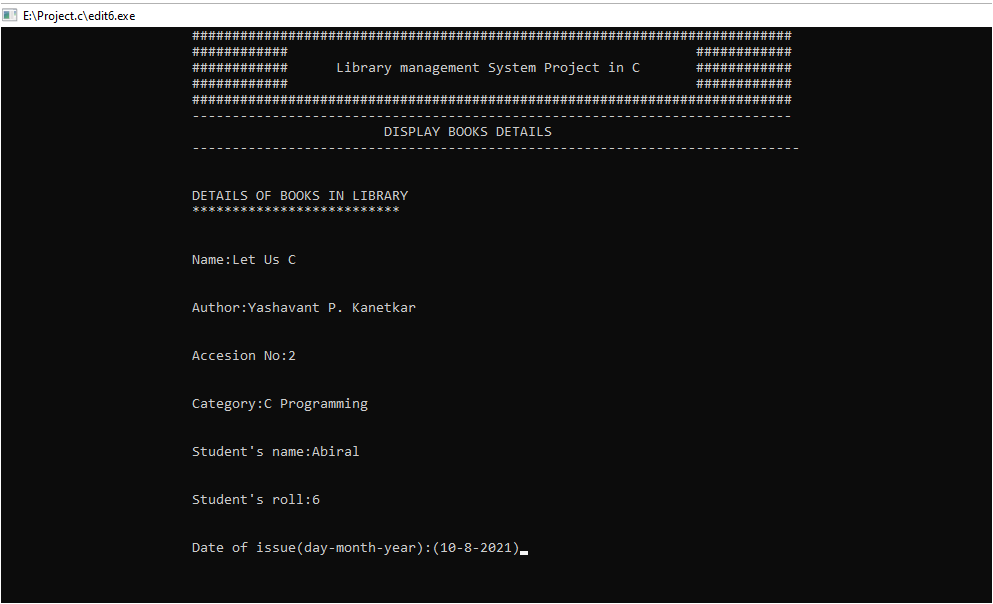
, Above, we can see the Main Menu where 7 different options are given, among which we have to choose one.

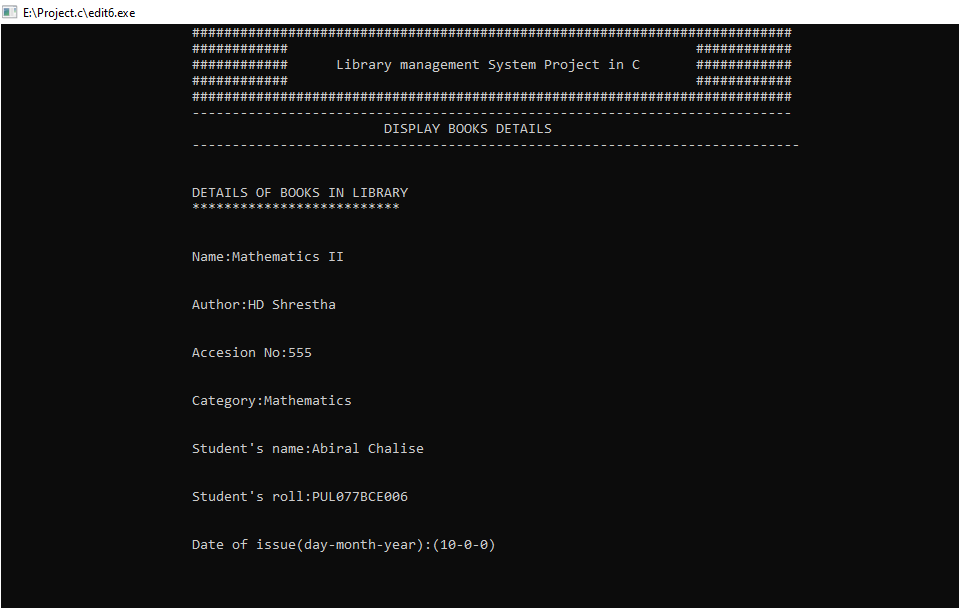


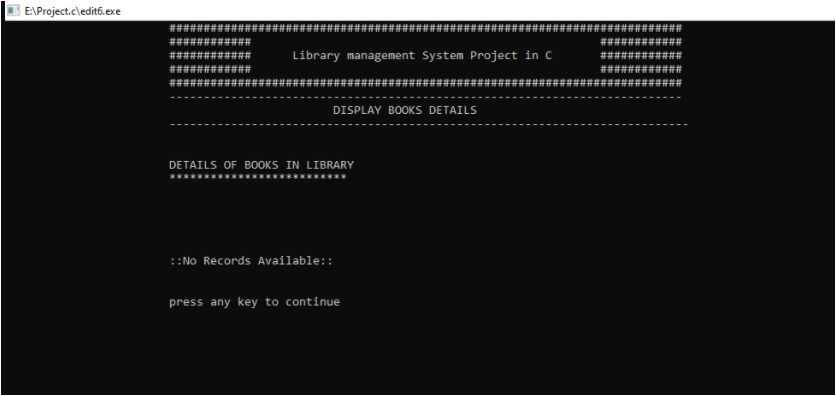
On pressing 1, we enter into the Add New Books section. Here, we have written the details of the books that we want to add in the record system of the library.



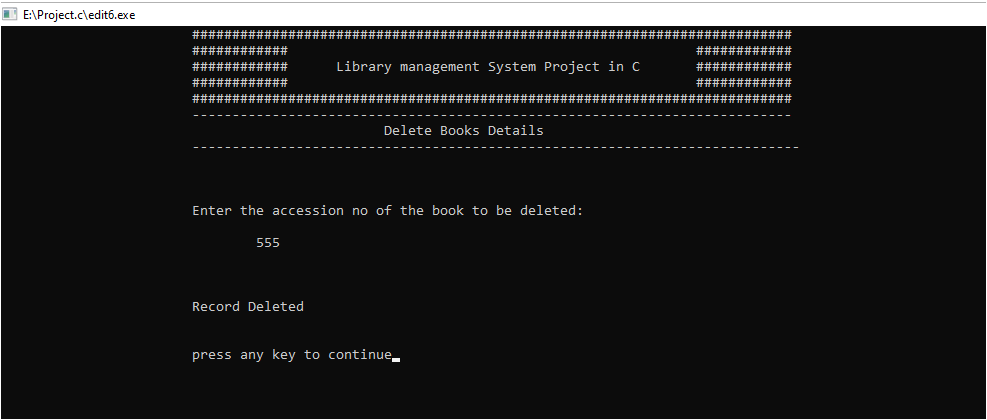
On pressing 2, we enter into the Display Books section. Here, it displays all the books that the students have borrowed from library.



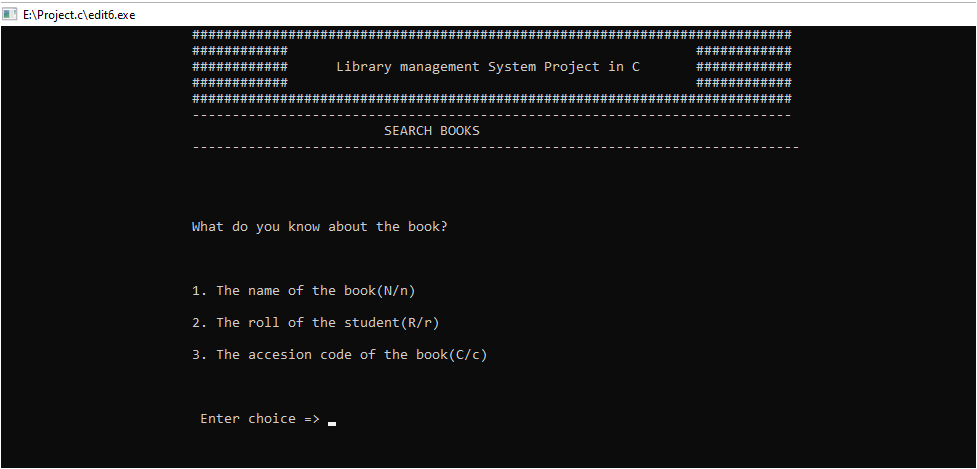




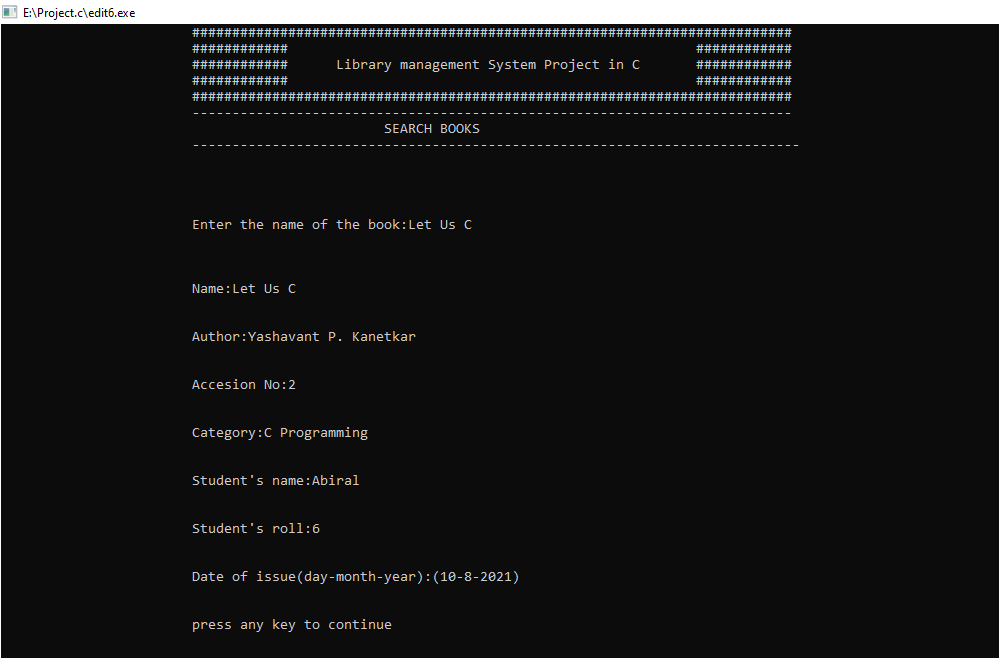
When the details of all the books are displayed, it displays a message that says there is ‘no more records available’.



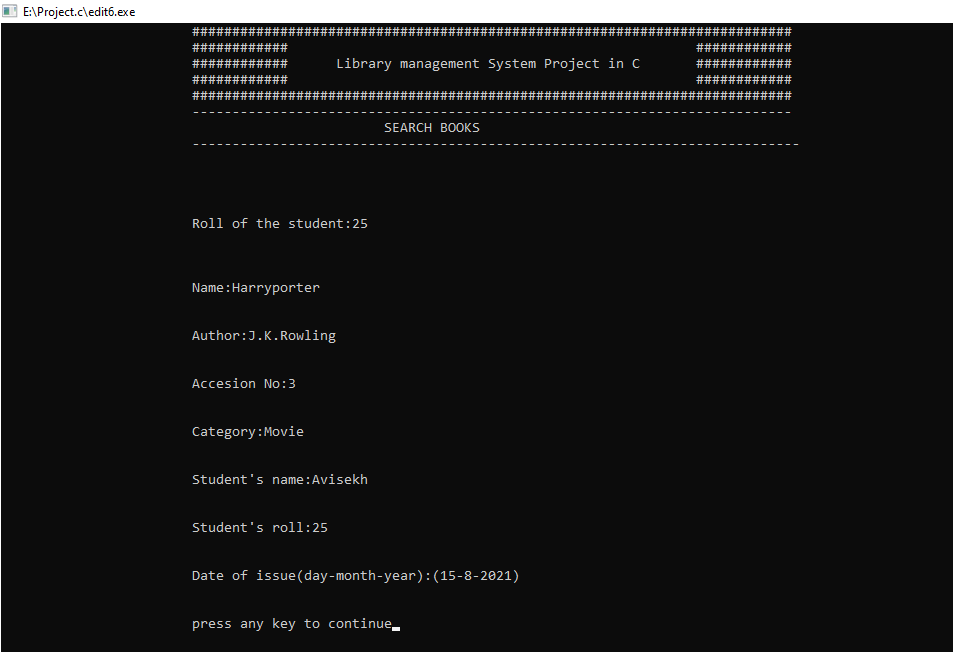
On pressing 3, we enter into the Delete Books section where the book will be deleted from the borrower’s list. When we type the accession code of the book that has been borrowed, the record will be deleted from the system.



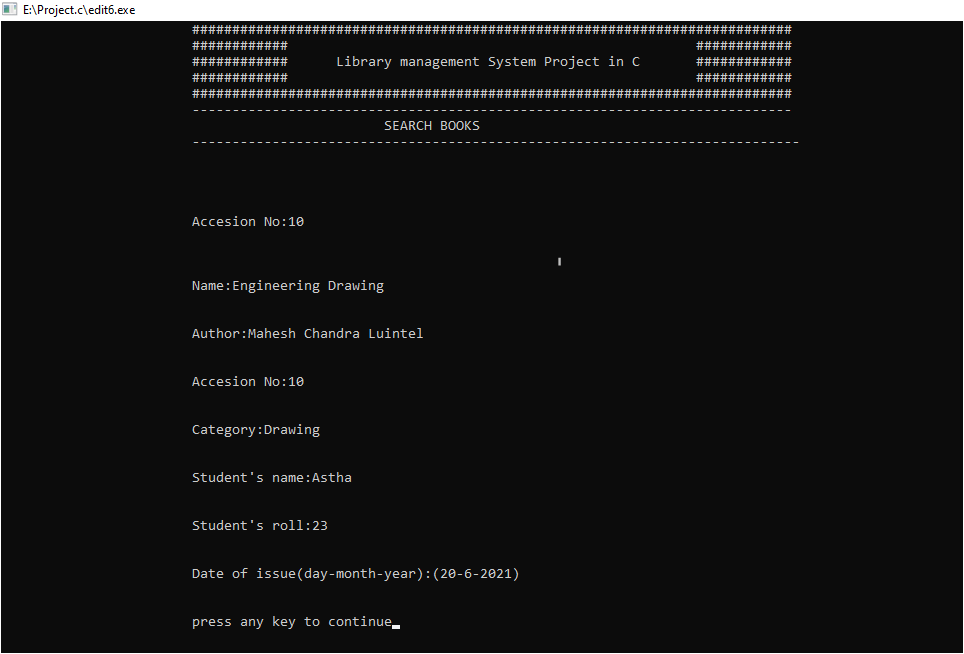
On pressing 4, we enter into the Search Books section. Further, the system asks us what we know about the book that we are searching in the library. We have to choose among the name of the book or the roll of the student who has borrowed the book or the accession code of the book.



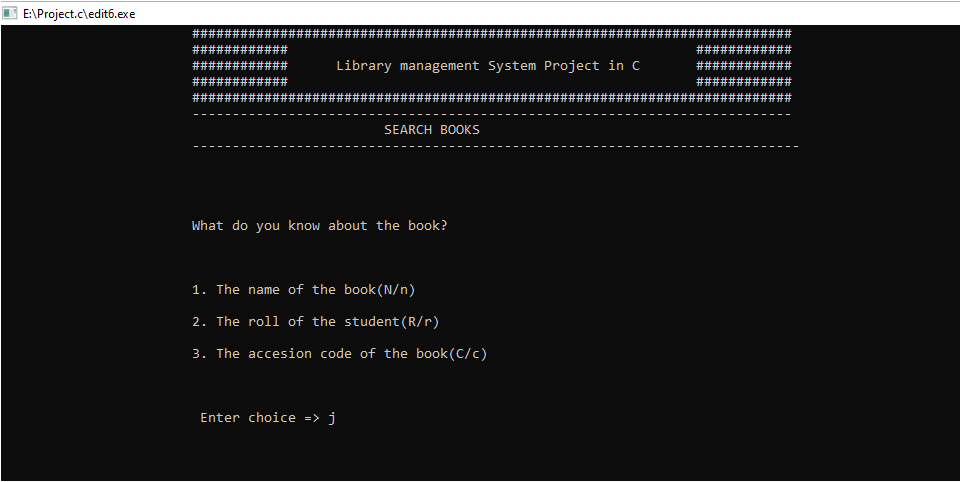
Here, as we can see, we can search the book in terms of the name. When we type the name of the book, all the details of the book are displayed.



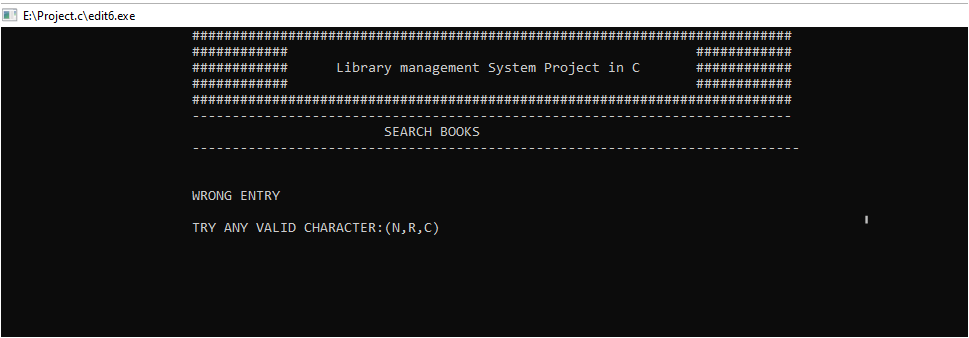
Similarly, here we can search the book in terms of the roll no. of the student who has borrowed the book from the library.



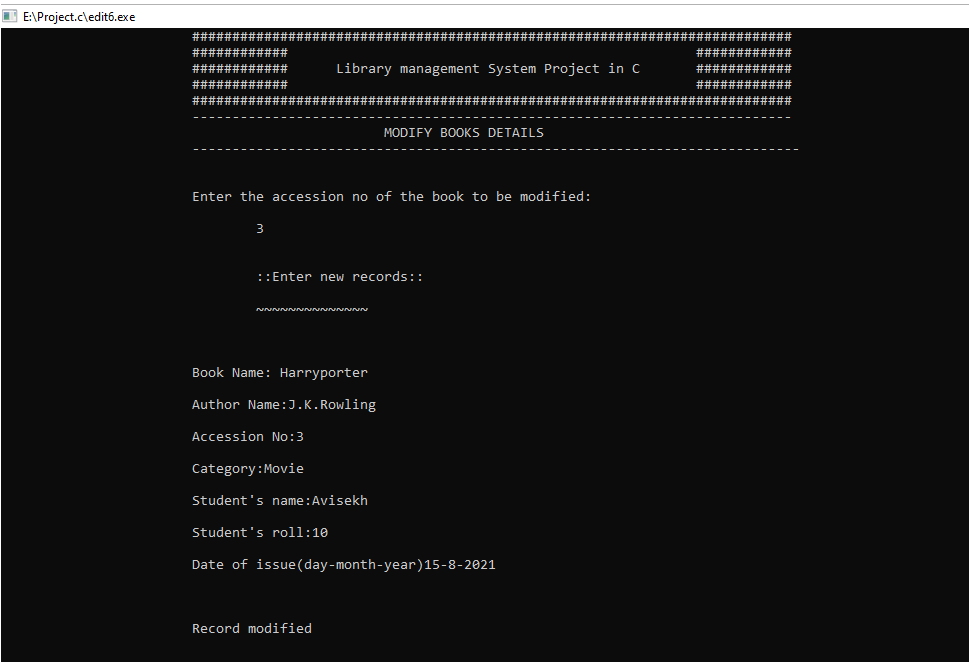
Finally, here we can search the book in terms of the accession no. of the book.



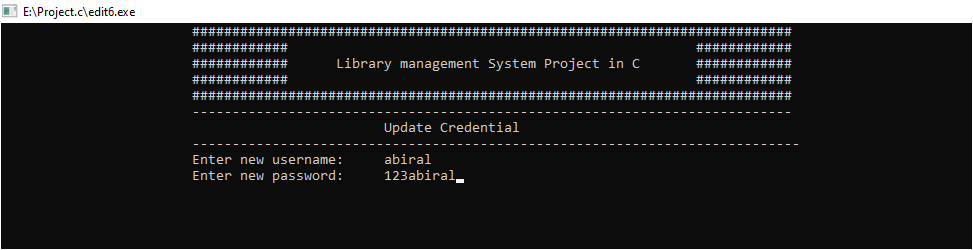
In this screenshot, we typed ‘j’ while we had to choose among N/n for name,R/r for roll or C/c for accession code.



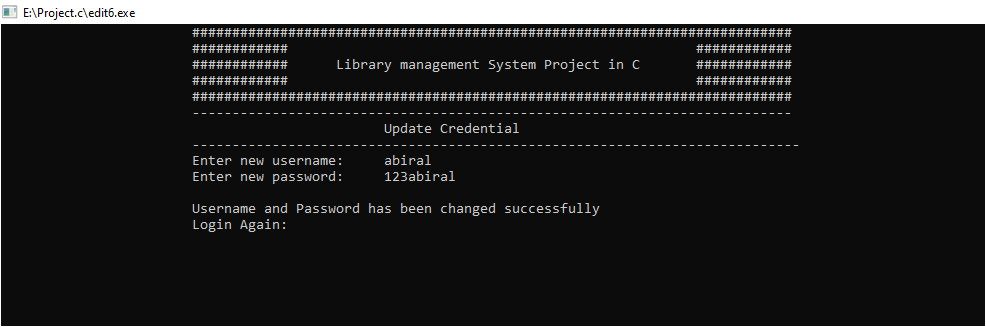
On typing ‘j’, a message was displayed on the screen that says ‘Wrong Entry’. We have to try the valid characters (N/n, R/r, C/c).



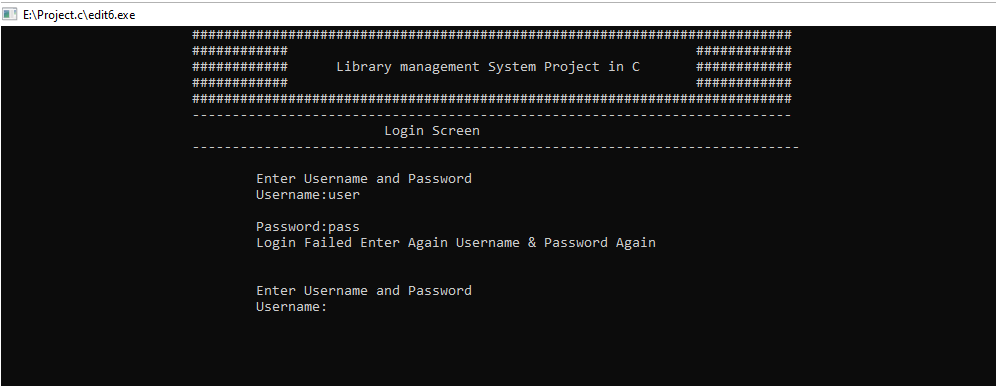
On pressing 5, we enter into the Modify Book section. It asks us the accession code of the book that is to be modified. Then, the new records are to be typed. After writing all the information that we want to change, we have to press enter, then, the record will be modified.



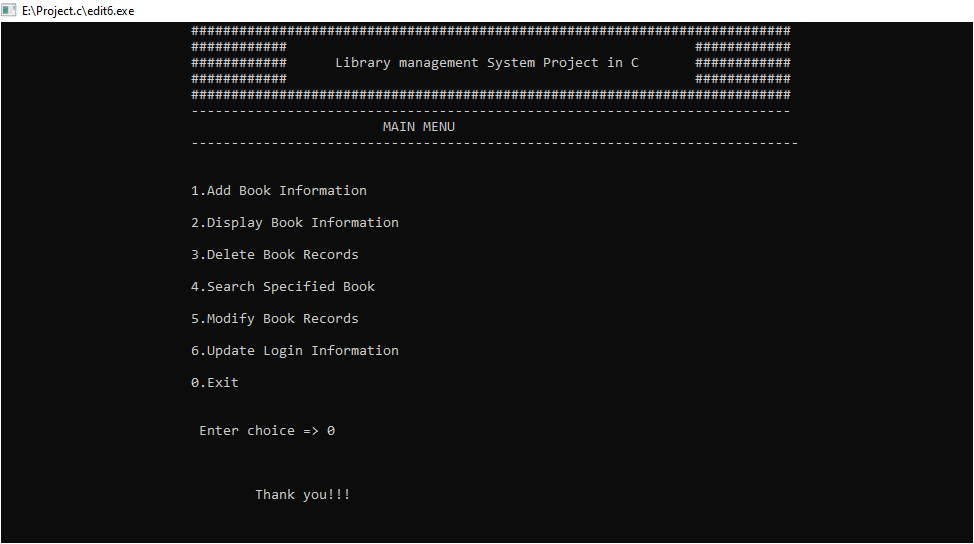
On pressing 6, we get into Update Credential section. Here, the username and password is updated.



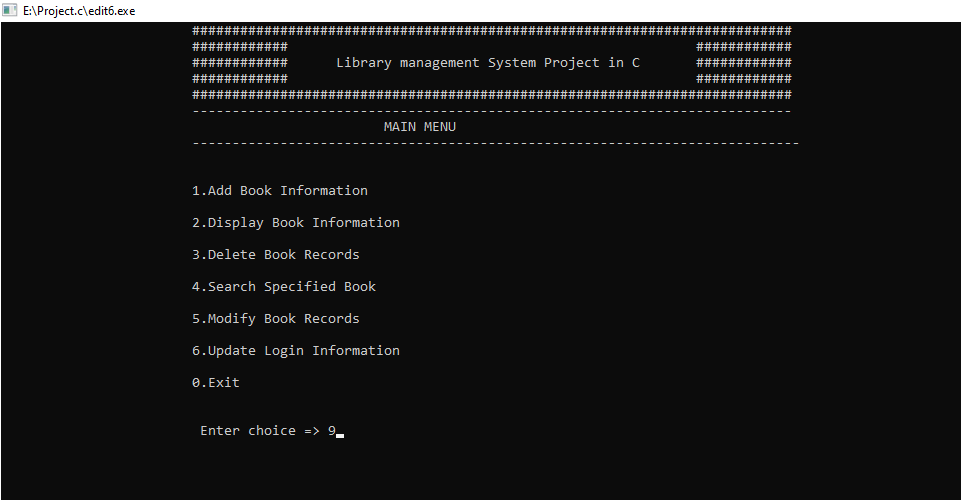
As we can see, after updating the username and password, the system asks us to login again.



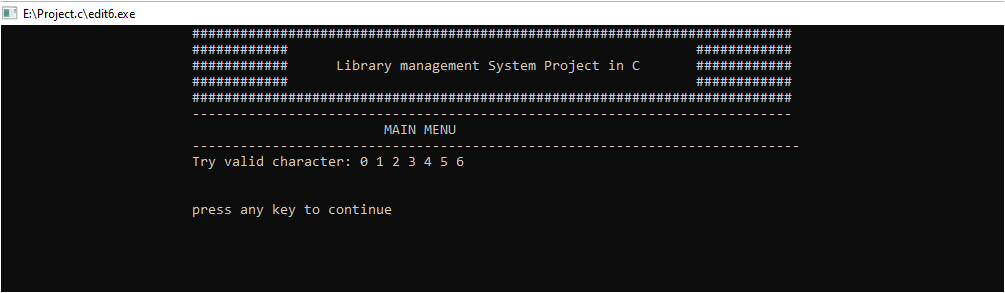
On entering the initial username and password, login is failed. To log into the system, we have to type the updated username and password.



On pressing 0, we exit from the system.



Getting back to the Main Menu, here, we have entered no 9.



On entering 9, a message is displayed that asks us to ‘try valid characters i.e. (0,1,2,3,4,5,6)’.

**DISCUSSION**

As the course study we studied C programming language in the 1st Semester of BCE. Knowing the fact that no course is complete only with the theoretical knowledge, we are assigned with the mini project on C programming that includes the entire topic that we covered during our study. With the same objective we have created a LIBRARY MANAGEMENT SYSTEM that carries the records of students who use the library. The program consists of the main concept of file handling.

At the starting of the program, we declared the entire header file that is used in the program that includes stdio.h, stdlib.h, string.h and conio.h. We have written our program in VS code as it is easier and convenient for programming. We used control statements, looping, array, pointer, structure, files etc. in this program. This program can add, display, delete, search and modify the book records. Firstly, there is a login screen where the user has to enter the username and password. Then, we get into the main menu where we can choose among 6 options which include: add book information, display book information, delete book information, search specified book, update login information or exit the program. The program then runs after any one option is selected by the user.

At first users are required to login in the system before entering the program. The default username and password for login are given user and pass respectively

**Add book information**: In the library, different books need to be borrowed by anyone. So, under the ‘add book information’ function, the details of the new book that is being borrowed by the student are to be entered. It includes name of the book, author’s name, accession no. of the book, category of the book (e.g.: math, physics, drawing, etc), student’s name, roll no, and the date of issue. We can add any number of books, after writing all these information, the system asks us to press esc if we want to exit, and again takes us back to the main menu.

**Display book information:** Under the ‘display book information’ function, the details of the book that the student has borrowed is shown. Whatever we write under the ‘add book information’ is displayed on entering number 2.

**Delete book information:** Underthe ‘delete book information’ function, the system asks us the accession no of the book that is to be deleted. Here, deletion is done after the borrowed book is returned to the library.

**Search specified book:** Under ‘search specified book’ function, the system asks us the information that about the issued book. It includes the name of the book(N/n), roll no of the student(R/r) and the accession code of the book(C/c). After giving any one of these information, the system tells us if the issued book is in the library records or not.

**Modify book records:** Under the ‘modify book records’ function, it asks us the existing accession code of the book to be modified. After entering with the accession code , the new records details are asked. It includes name of the book, author’s name, accession no. of the book, category of the book (e.g.: math, physics, drawing, etc) student’s name, roll no, and the date of issue. Pressing enter, the records will be modified.

**Update login information:** Under the ‘update login information’ function, the new username and password are asked. After we reset them, the system stops working with the older username and password. To get logged in, we have to type the new username and password.

**Exit:** On pressing 0, the system is exited.

In the program, we have used fflush(stdin) frequently. The use of fflush(stdin) is to flush the buffer region. While performing the programming, the RAM and the hard disk have to process spontaneously but since the processing speed of  the different standard input device is not same so a region in the RAM  is created which stores the data stored in the file temporarily. This region in the RAM is called buffer which needed to be flushed time and again for which we used fflush(stdin).

During program compilation, we faced different problems that needed to be handled. In this program, some logical errors were detected in deleting, modifying and searching operations .For debugging these logical errors, we took help of teachers and the seniors. By facing these difficulties we got the knowledge of new facts such as how and where proper use of different mode in file handling is done. During compilation also we solved the problem by group discussion.

In this way we successfully completed our project.

**SUMMARY**

Our main aim in this program is to prepare issued book records in library. In our opinion,adding book information, displaying book information, deleting book records, searching for a specified book, modifying book records, updating login information and exiting are the most basic categories of any library . So, we have included them. The class modules and problems through done on the computer itself were much theoretical because we were unable to understand their application in our daily life.

The mini project on the other hand not only taught us practical use of program but also helped us in gaining skills like co-ordination, leadership, management of time, planning, foresights, etc. Above all, it taught us that we could do something useful with C-programming. It gave taste of being a programmer which will surely be very useful in our future.

Thus we can predict that we succeeded on our aim. The problems we faced make us more confident in C language.

**REFERENCE**

* Learning C by examples                                          Er. Krishna Kandel
* Course Note Mr. Santosh Giri
* Programming in ANSI C Bala Guru Swami
* Let Us C Yashavant P. Kanetkar
* A textbook of C Programming Baburam Dawadi/Ram Dutta Bhatta