|  |  |
| --- | --- |
| Computer science unit 1 internal assessment | **Candidate Name: Jenalyn C. Weekes**  **Candidate Number:1102010585**  **Subject: Computer Science Unit 1**  **Subject Teacher: Ms. Donna Ible-White**  **School: Montserrat Community College**  **Centre Number: 110201** |

Contents

[**PROBLEM DEFINITION** 2](#_Toc513050940)

[**ALGORITHM** 3](#_Toc513050941)

[**NARATIVE** 3](#_Toc513050942)

[**PSEUDOCODE** 11](#_Toc513050943)

[**PROGRAM CODE** 19](#_Toc513050944)

[**TEST PLAN** 31](#_Toc513050945)

[**SCREENSHOTS** 33](#_Toc513050946)

[**BIBLIOGRAPHY** 36](#_Toc513050947)

# **PROBLEM DEFINITION**

The Montserrat Community College (MCC), located in Salem, Montserrat, began operating in September of 2004. The school has approximately 55 students enrolled, not including part-time students. The school offers CXC CAPE Associate Degree programmes where the students may pursue a course of their choice over the course of two to five years. The courses available include Accounting, Biology and Digital Media. A part of the school’s vision is to “… provide expanded access to quality education and training programmes, which will satisfy the personal and professional aspirations of the people of Montserrat…”

The school’s learning resource center or library is open to all students, part-time and full-time alike. However, the library does most of its operations manually, where most of the information is written on paper. The problem with this is that often times, the paper is misplaced and it becomes almost impossible to determine who borrowed which book and when. Additionally, the librarian is not able to provide a complete list of books available at the library. Instead, she has to resort to physically going to the book shelves each time to display the books available on a particular topic or subject. This outdated system is not only inefficient but also very unreliable.

In order to solve this problem, a program will be created to properly manage the library. The program should be able to save information for all of the books which includes the name of the book, author of the book and book id. This information will be stored to a file for quick and reliable retrieval. Furthermore, a log should be created to track books that are borrowed from the library.

# **ALGORITHM**

## **NARATIVE**

**START**

Declare numbk, cnt, counter, choice, cat, num, i as integers

Declare password, username, bk, as character arrays

Declare books as struct with character variables Bookname, Bookauthor and cat and integer variable BoodID

Create instance of books struct

***Password Function***

START

Create struct with ‘Username’ and ‘Pass’ as character arrays

Create instances ‘user1’, ‘user2’ and ‘user3’ of the struct user

Assign “Jane” and “passworda” to Usernames and Pass respectively in user1

Assign “John” and “passwordb” to Usernames and Pass respectively in user2

Assign “James” and “passwordc” to Usernames and Pass respectively in user3

For 3 attempts

Prompt user for their username

Store in ‘username’

Prompt user for password

Store in ‘password’

Compare username to the Usernames of user1, user2 and user3

Compare password to Pass of user1, user2 and user3

If username is equal to Username and password is equal to Pass for user1

Attempts equals 5

Else if username is equal to Username and password is equal to Pass for user2

Attempts equals 5

Else if username is equal to Username and password is equal to Pass for user3

Attempts equals 5

Otherwise prompt the user to try again

If cnt is equal to 3

Call Quit function

Else Call Menu function

STOP

***AddBook Function***

START

Declare c as an integer

Prompt user for the number of books that they would like to add

Read numbks

If numbks is not an integer

Output “You must not enter an integer. How many books would you like to add?”

Otherwise Output “Adding *numbks* Book(s) ”

For numbks repetition do

Open file ‘Library.bks.txt’ Prompt user to enter the number of book they would like to add

Output “Select a category: 1.Mathematics 2.English 3.Fiction 4.Non-Fiction 5.Return To Main Menu”

Read cat

If cat is equal to 5

Call Menu function

Return to Menu Function

Output “Enter the information below”

Output “Category: ”

Read bk

Output “Book ID:”

Read BookID

Output “Book name: ”

Read Bookname

Output “Book Author: ”

Read Bookauthor

Write bk, BookID, Bookname, Bookauthor to file “library.bks.txt”

Close file “library.bks.txt”

STOP

***Menu Function***

START

Declare c as an integer

For 5 attempts do:

Output “Library Main Menu”

Output “Add New Books……....1”

Output “Borrow Book………….2”

Output “Display Books…………3”

Output “Exit…………………….4”

Store input to choice

While user input choice is not an integer, do

Output “You must enter an integer.”

Output “Library Main Menu”

Output “Add New Books………..1”

Output “Borrow Book………….2”

Output “Display Books…………3”

Output “Exit…………………….4”

If choice equal 1

Send user to AddBook Function

If choice equal 2

Send user to BorrowBookLog Function

If choice equal 3

Send user to Display Function

Call Menu function

If choice equal 4

Send user to Quit Function

If choice does not equal 1, 2, 3 or 4 prompt user to try again

STOP

***Quit Function***

START

Output “\*\*\*\*\*Have a nice day!\*\*\*\*\*”

Exit program

STOP

***Display Function***

START

Assign character variable c to the first character of the file ‘Library.bks.txt’

Assign character variable eof to the last character of the file ‘Library.bks.txt’

Open file ‘Library.bks.txt’ to be read

If the file pointer is not returned

Output “Error! File does not exist”

Otherwise print contents of file

Close file ‘Library.bks.txt’

STOP

***BorrowBookLog Function***

START

Declare numofbks and loopcounter as integers

Declare category as an array of characters

Output “How many books would you like to borrow?”

Store user input in numofbks

Output “BORROWING *numofbks* BOOK(S)”

For numofbks repetitions

Output “Books available: ”

Call Display function

Declare c as an integer

Open file ‘Library.borrowedbks.txt

Output “Please enter information for the book you would like to borrow .”

Output “Category: ”

Read category

Output “Book ID: ”

Read BookID

Output “Book name: ”

Read Bookname

Output “Book Author: ”

Read Bookauthor

Declare struct tm to get values for the date

Declare date as character array

Assign values for the actual date to variable date

Write category, BookID, Bookname, Bookauthor, date to file ‘Library.borrowedbks.txt’

Output “BOOK SUCCESSFULLY BORROWED”

Close file ‘Library.borrowedbks.txt’

Call Menu function

STOP

***Main Function***

START

Call the password function

STOP

**STOP**

## **PSEUDOCODE**

**START**

DECLARE INT choice, numbks, cnt and cat

ASSIGN INT counter = 0

DECLARE STRING Pass, UserName

DECLARE bk as a character array with values "Mathematics", "English", "Fiction", "Non-Fiction"

CREATE FILE Library.bks

DECLARE Password ()

DECLARE AddBooks ()

DECLARE Menu ()

DECLARE Quit ()

DECLARE BorrowedBookLog ()

***VOID Password ()***

START

DECLARE STRUCT user

DECLARE char array UserNames

DECLARE char array Pass

FOR (cnt = 1 to 3)

PRINT “PLEASE ENTER YOUR USERNAME”

READ UserName

PRINT “PLEAS ENTER YOUR PASSWORD”

READ Pass

IF UserNames AND Pass = UserNames AND Pass for user 1

cnt = 5

ELSE IF UserNames AND Pass = UserNames AND Pass for user 2

cnt = 5

ELSE IF UserNames AND Pass = UserNames AND Pass for user 3

cnt = 5

ELSE

PRINT “Access Denied. Try again.”

END FOR

IF CNT =3

CALL Quit ()

END IF

ELSE

Menu ()

STOP

***END Password ()***

***VOID AddBook ()***

START

DECLARE INT c

PRINT “How many books would you like to add?”

READ numbks

PRINT “ADDING numbks BOOK(S)”

WHILE counter not = numbks

PRINT “Select a category: 1.Mathematics 2.English 3. Fiction 4.Non-fiction 5.Return to Main Menu”

READ cat

IF cat =5

CALL Menu ()

END IF

OPEN ‘Library.bks.txt’

PRINT “Enter the information below.”

PRINT “Book ID: ”

READ num

PRINT “Book Name: ”

READ Bookname

PRINT “Book Author: ”

READ Bookauthor

CLOSE ‘Library.bks.txt’

PRINT “BOOK SUCCESSFULLY ADDEDD”

END WHILE

STOP

END Addbooks ()

***DECLARE Menu ()***

***VOID Menu ()***

START

WHILE choice > 5 OR choice < 1

PRINT “LIBRARY MENU”

PRINT “Add New Book………………………..1”

PRINT “Remove Book…………………………2”

PRINT “Borrow Book………………………….3”

PRINT “Exit…………………………………….4”

READ choice

IF choice == 1

CALL AddBook ()

IF choice ==2

CALL BorrowingBookLog ()

IF choice ==3

CALL Display ()

CALL Menu ()

IF choice ==4

CALL Quit ()

ELSE

PRINT “Something went wrong, please try again.”

END WHILE

STOP

***END Menu ()***

***INT Quit ()***

PRINT “Have a nice day”

EXIT

***END Quit ()***

***INT Display ()***

OPEN ‘Library.bks.txt’

DECLARE STRING c = first character of ‘Library.bks’

IF ‘Library.bks.txt’ = null

Print “Error! File does not exist.”

END IF

ELSE

WHILE true

C = file pointer

IF end of file = file pointer

RETURN false

END IF

PRINT c

END WHILE

RETURN 0

***END Display ()***

***VOID BorrowBookLog ()***

DECLARE INT numofbks, loopcounter

DECLARE STRING category

PRINT “How many books would you like to borrow?”

READ numofbks

FOR 0 < loopcounter < numofbks

Display ()

DECLARE INT c

OPEN ‘Library.borrowedbks.txt’

PRINT “Please enter information for the book you would like to borrow”

PRINT “Book ID: ”

READ num

PRINT “Book Name: ”

READ Bookname

PRINT “Book Author: ”

READ Bookauthor

DECLARE array date

ASSIGN date = day/month/year

PRINTcategory, num, Bookname, Bookauthor, date into ‘Library.borrowedbks’

PRINT “BOOK SUCCESSFULLY BORROWED”

CLOSE ‘Library.borrowedbks.txt’

END FOR

CALL Menu ()

***END BorrowBookLog ()***

***INT Main ()***

CALL Password ()

IF Password () returns 1

***END Main ()***

**STOP**

# **PROGRAM CODE**

//////////////////////////////////////////////////////////////////

// Library Management Program //

// Program: Library Management //

// Candidate's Name: Jenalyn Weekes //

// Candidate's Number: 1102010585 //

// Centre Number: 110201 //

// Class: Computer Science Unit 1 //

// Territory: Montserrat //

// Lecturer: Miss Donna Ible-White //

// //

// Project: Internal Assessment (IA) //

// This program serves to manage books in a library system. //

//////////////////////////////////////////////////////////////////

#include<stdio.h>

#include <time.h>

#include<conio.h>

#include<string.h>

#include <stdlib.h>

#include <ctype.h>

#define MAXINPUT 100

void Menu();

void BorrowBookLog();

int Display();

int numbks, cnt, counter=0;

int choice;

char password[10], username[15];

int cat;

int num, i=0;

char bk[][20]= {"Mathematics", "English", "Fiction", "Non-Fiction"};

char s[15];

FILE \*fp;

FILE \*ptr;

struct books //declares struct book with variables Bookname, Bookauthor,BookID, \*cat and Status

{

char Bookname[20];

char Bookauthor[20];

int BookID;

char \*cat;

};

struct books a;

void Password()//This function allows users to login to the library management system.

{

struct user //creates struct 'user'

{

char UserNames[15];

char Pass[10];

};

struct user user1={ //creates intance of struct 'user1'

"Jane",

"passworda"

};

struct user user2={

"John",

"passwordb"

};

struct user user3={

"James",

"passwordc"

};

for(cnt=1; cnt<=3; cnt++)

{

printf("Welcome... \n\nPlease enter your user name:"); //prompts user for user name

scanf("%s", username); //reads username

printf("Please enter your password:");//prompts user for password

scanf("%s", password); //reads password

if (strcmp(username, user1.UserNames) == 0 && strcmp(password, user1.Pass) == 0) //checks whether the username and password match that of struct user1

cnt = 5; //cnt assigned value 5 to leave loop

else if (strcmp(username, user2.UserNames) == 0 && strcmp(password, user2.Pass) == 0)//checks whether the username and password match that of struct user2

cnt = 5; //cnt assigned value 5 to leave loop

else if (strcmp(username, user3.UserNames) == 0 && strcmp(password, user3.Pass) == 0)//checks whether the username and password match that of struct user3

cnt = 5; //cnt assigned value 5 to leave loop

else //if password and username are not a match user is prompted to try again

printf("\nAccess Denied.\nTry Again\n");

system("pause");

system("cls");

}

if (cnt==3){

Quit();

}

else{

Menu();

}

system("cls");

}

void AddBook()//This function allows user to add books to the library

{

int c;

printf("How many books would you like to add? ");

while (scanf("%d", &numbks) != 1) //loops while the value entered for 'numbks' is not an integer

{

while (!isspace(c = getchar())); //loops while the value for c is not a blank space

{

system("cls");

printf("You must enter an integer.");

printf("\nHow many books would you like to add? ");

}

}

printf("\nADDING %d BOOK(S)\n", numbks);

do

{

int c;

printf("Select a category: \n1.Mathematics \n2.English \n3.Fiction \n4.Non-Fiction \n5.Return To Main Menu"); //prompts user to choose a category

while (scanf("%d", &cat) != 1) //loops while the value entered for 'cat' is not an integer

{

while (!isspace(c = getchar()));//loops while the value for c is not a blank space

{

system("cls");

printf("You must enter an integer.\n");

printf("Select a category: \n1.Mathematics \n2.English \n3.Fiction \n4.Non-Fiction \n5.Return To Main Menu");

}

}

if(cat==5) //checks whether user would like to return to the menu

{

system("cls");

Menu() ;

}

fflush(stdin);

system("cls");

fp=fopen("Library.bks.txt","a+"); //opens file "Library.txt" to be appended

a.cat=bk[cat-1];

printf("Enter the Information Below.\n");

printf("Category: ");

printf("%s",bk[cat-1]);

printf("\nBook ID: ");

while (scanf("%d", &num) != 1) //loops while the value entered for 'num' is not an integer

{

while (!isspace(c = getchar()));

{

system("cls");

printf("You must enter an integer.\nBook ID: ");

}

}

fflush(stdin);

a.BookID=num;;

printf("Book Name:");

scanf("%s", &a.Bookname);

fflush(stdin);

printf("Book Author:");

scanf("%s", &a.Bookauthor);

fflush(stdin);

fprintf(fp, "Category:%s. Book ID:%d. Book Name:%s Book Author:%s \n", bk[cat-1], a.BookID, a.Bookname, a.Bookauthor);//adds book information to file

fclose(fp);

system("cls");

printf("BOOK SUCCESSFULLY ADDED\n");

counter++;

}

while(counter!=numbks);//loops through once the counter is not equal to the number of books the person wants to add

Menu();

}

void Menu()//This function directs users to a menu with options to choose from

{

int c;

do

{

printf("Library Main Menu \n");

printf("Add New Book..........1\n");

printf("Borrow Book...........2\n");

printf("Display Books.........3\n");

printf("Exit..................4\n");

while (scanf("%d", &choice) != 1) //loops while the value entered for 'choice' is not an integer

{

while (!isspace(c = getchar()));//loops while the value for c is not a blank space

{

system("cls");

printf("You must enter an integer.\n\n");

printf("Library Main Menu \n");

printf("Add Books..........1\n");

printf("Borrow Books...........2\n");

printf("Display Books.........3\n");

printf("Exit..................4\n");

}

}

switch(choice)

{

case 1://if choice is 1 takes user to AddBooks function

{

system("cls");

AddBook();

break;

}

case 2://if choice is 2 takes user to BorrowBookLog function

{

system("cls");

BorrowBookLog();

break;

}

case 3://if choice is 3 takes user to Display function

{

system("cls");

Display();

system("pause");

system("cls");

Menu();

break;

}

case 4://if choice is 4 takes user to Quit function

{

system("cls");

Quit();

break;

}

default://if choice is not 1,2,3 or 4 the following is carried out

system("cls");

printf("Something went wrong, please try again.\n");

}

}

while(choice > 5|| choice < 1); // performs do while choice is less than or equal to 5

}

int Quit()//This function exits the program

{

printf("\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Have a nice day!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");

exit (0);

system("cls");

}

int Display()//This function displays books in library

{

char c = fseek(fp, 0, SEEK\_SET); //assigns the variable c to the first character of the file

char eof = fseek(fp, sizeof("library.bks.txt"), SEEK\_END);

fp = fopen("library.bks.txt","r"); //opens file "Library.txt" to be read

if(fp == NULL) //if the file pointer is not returned, the user is informed that the file does not exist

{

printf("Error! File does not exist.\n");

system("pause");

system("cls");

}

else //prints contents of file

while(1)

{

c = fgetc(fp);

if(feof(fp))

{

return 0;

break;

}

printf("%c", c);

fflush(stdout);

}

fclose(fp);

return 0;

}

void BorrowBookLog()//This function tracks books borrowed by customers

{

int numofbks, loopcounter;

char category[30];

int c;

printf("How many books would you like to borrow?");

while (scanf("%d", &numofbks) != 1) //loops while the value entered for 'num' is not an integer

{

while (!isspace(c = getchar()));

{

system("cls");

printf("You must enter an integer.\nHow many books would you like to borrow?\n");

}

}

printf("\nBORROWING %d BOOK(S)\n", numofbks);

for(loopcounter=0; loopcounter<numofbks; loopcounter++)

{

printf("Books available: \n\n");

Display();

int c;

ptr=fopen("Library.borrowedbks.txt","a+"); //opens file "borrowedbks.txt" to be appended

a.cat=bk[cat-1];

printf("\n\nPlease enter information for the book you would like to borrow .\n");

printf("\nCategory: ");

scanf("%s", &category);

printf("Book ID: ");

while (scanf("%d", &num) != 1) //loops while the value entered for 'num' is not an integer

{

while (!isspace(c = getchar()));

{

system("cls");

printf("You must enter an integer.\nBook ID: ");

}

}

fflush(stdin);

a.BookID=num;;

printf("Book Name:");

scanf("%s", &a.Bookname);

fflush(stdin);

printf("Book Author:");

scanf("%s", &a.Bookauthor);

fflush(stdin);

//Gets the values for the current date

struct tm \*tm;

time\_t t;

char date[100];

t = time(NULL);

tm = localtime(&t);

strftime(date, sizeof(date), "%d/%m/%Y", tm);

printf("%s",date);

fprintf(ptr, "Category:%s. Book ID:%d. Book Name:%s Book Author:%s Date:%s\n", category, a.BookID, a.Bookname, a.Bookauthor, date);//adds book information to file

printf("\nBOOK SUCCESSFULLY BORROWED\n");

fclose(ptr);

system("pause");

system("cls");

}

Menu ();

}

int main()

{

Password();

return 0;

}

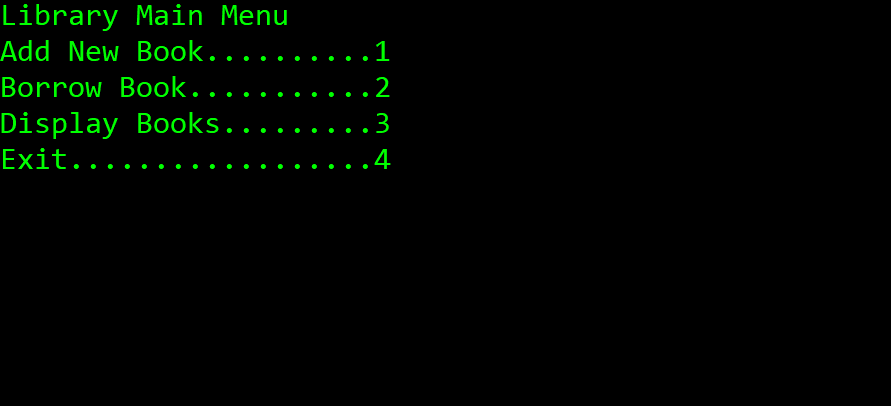
# **TEST PLAN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Plan** | **Description** | **Expected Result** | **Actual Result** | **Success/**  **Comments** |
| Password Function | Normal | User entered “Jane” as username and “passworda” as password | Program should direct user to the main menu. | “Press any key to continue” | Test case was successful |
| Extreme | User entered “JWeekes” as the username and “password” as the password. | Program should display error message that combination is incorrect | “Access Denied. Try again.” | Test case was successful |
| Erroneous | User entered “@@@`” as the username and “1234$” as the password. | Program should display error message that combination is incorrect | “Access Denied. Try again.” | Test case was successful |
|  |  |  |  |  |  |
| Menu Function | Normal | User entered “1” | Program should direct user to the add books function. | “How many books would you like to add.” | Test case was successful |
| Extreme | User entered “7” | Program should display error message. | “Something went wrong, please try again.” | Test case was successful |
| Erroneous | User entered “abc” | Program should display error message. | “You must enter an integer.” | Test case was successful |
|  |  |  |  |  |  |
| Add Books Function | Normal | User entered “1” | Program should direct user to the add books function. | “How many books would you like to add.” | Test case was successful |
| Extreme | User entered “7” | Program should display error message. | “Something went wrong, please try again.” | Test case was successful |
| Erroneous | User entered “a” | Program should display error message. | “You must enter an integer.” | Test case was successful |
|  |  |  |  |  |  |
| Display Books Function | Normal | User selects display option when the file has books. | Program should show data in ‘Library.bks.txt’ file’ | Program showed data in ‘Library.bks.txt’ file’ | Test case was successful |
| Extreme | User selects display option when the file has no books. | Program should display message that the file is empty and return to Menu. | Program returned to Menu | Test was partially successful as it brought the user to Menu but failed to show error message. |
|  |  |  |  |  |  |
| BorrowBookLog Function | Normal | User enters the “1” | Program should continue in the BorrowBoogLog function. | Program returned “BORROWING 1 BOOK(S)” | Test case was successful |
| Erroneous | User enters “abc” | Program should display error message. | Program returned “You must enter an integer.” | Test case was successful |

# **SCREENSHOTS**

Figure showing a successful login

Figure showing display of Menu function



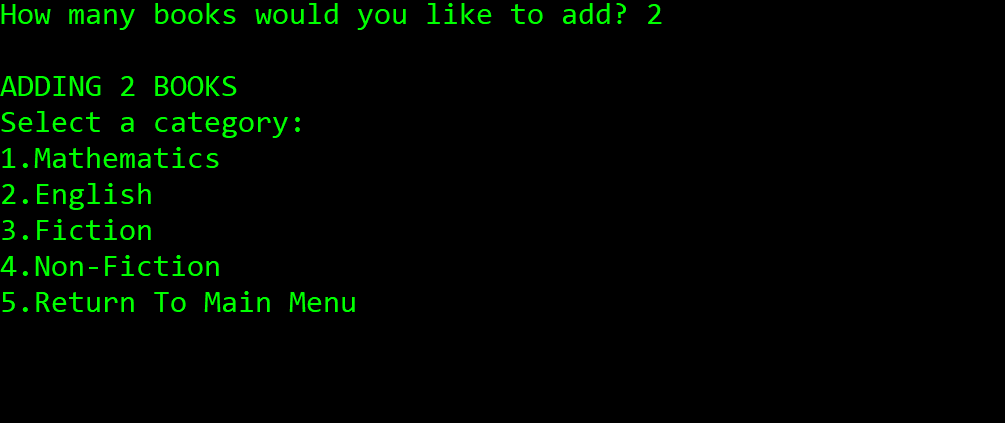


Figure showing user adding books to library system.

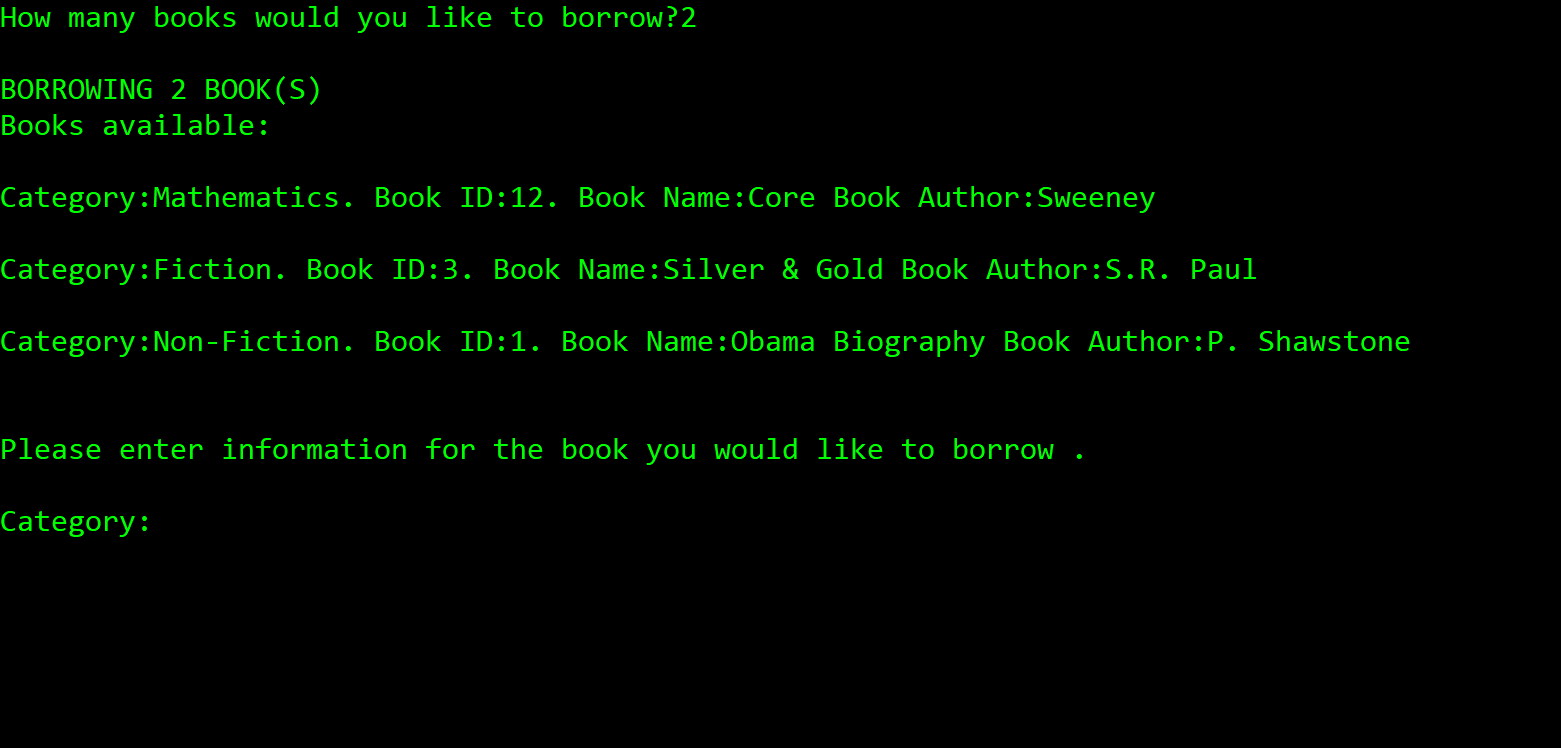
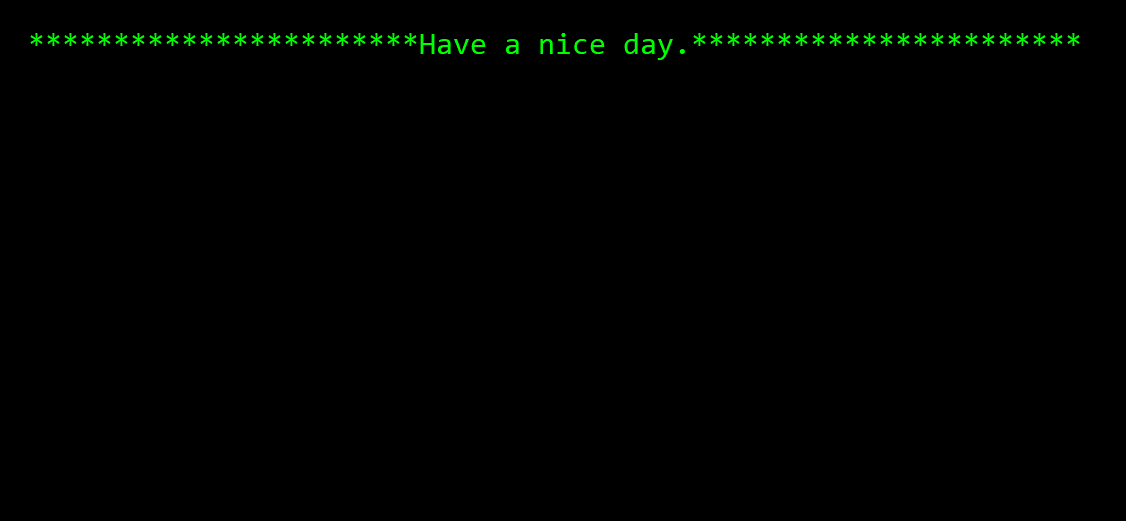


Figure showing user borrowing books from the library system.

Figure showing exit screen as the program is terminated.



# **BIBLIOGRAPHY**

P, D., & H, D. (2013). *How to Program with C.* Pearson Education, Inc.

West, J.-O. (2013). *Computer Science for CAPE Examinations: Unit 1.* Kingston.