Assignment 8

Library System Report

Group 8

Leader: Junming.Zhang 1301058

Members: Lanxiang.Wang 1302815

Jingchun.Xiang 1302523

Yizhi.Shi 1302203

Minghao.He 1302579

Assignment8

Question: Project B

The library of ABC town needs a new electronic rental system and your team is employed to build it.

Customer specifications:

•The implemented system should be able to handle the basic operations of a university library including the following features:

•It must be able to store the set of books including their author(s), title, ISBN, subject, loan type (normal, short loan, no-take-out), shelf-mark, loan status, number of copies, etc. Also, it should provide options for altering the details of a book, mark it as lost / damaged / ordered, etc.

•Registration of a new library user as teacher or student with varying privileges (e.g., teacher can keep books for more days), which can be altered for any existing user at any time.

•Search for a particular book and allocate a book to a specific borrower for a specific time period. Also, implement returns and renewals.

•It prints the list of available books and borrowed books.

•Provide borrower statistics (e.g. about the number of books) to a particular account.

Software Development Process

1. Problem statement:

•This exercise asks students to construct a university library system, which contains several main functions, namely:

* Register function
* Log in function
* Add books function
* Delete books function
* Modify books information function
* Borrow books function
* Return books function
* View available books function
* View on loan book function

* Delete Users function
* Search book function
* Personal information function

2. Analysis:

Register function

• input:

– the string entered by user (user name and user password)

• output:

– 2 files (text file and binary file)

• additional requirements or constraints:

– none

Log in function

• input:

– the string entered by user (user name and user password)

• output:

– login successful to the menu

• additional requirements or constraints:

– none

Add book function

• input:

– the string entered by user (ISBN, book name, author, and so on)

• output:

– 2 files (text file and binary file)

• additional requirements or constraints:

– none

Delete book function

• input:

– chose the book information to delete the string that entered by user (ISBN, book name, author, and so on)

• output:

– delete the book information

• additional requirements or constraints:

– none

Search book function

• input:

– the book information (ISBN, book name, author, and so on)

• output:

– the information of the book

• additional requirements or constraints:

– none

Delete User function

• input:

– chose the user information that the string entered by user (name, password, information)

• output:

– delete the user information

• additional requirements or constraints:

– none

Search book function

• input:

– the book information (ISBN, book name, author, and so on)

• output:

– the information of the book

• additional requirements or constraints:

– none

Borrow book function information

• input:

– the available books string that entered by user(enter book information)

• output:

– record the successfully borrowed books in files

• additional requirements or constraints:

– none

Return book function information

• input:

– the returned books string that entered by user(enter book information)

• output:

– record the successfully returned books in files

• additional requirements or constraints:

– none

View on loan book function

• read and display the borrowed books from files

View available books function

• read and display the available books from files

3. Design

First, it is the statement of the 'struct'. There are two structures, one is the structure of the book, there are all kinds of book information, add the last pointer in the structure to build a linked list. Another link list built by 'struct' is about the account. "Typedef" plays the role of the structure (book and account information) to be defined as a kind of new type (convenient for later program, can directly state book A, A is a 'struct', no more need the 'struct' statement)

After the two link lists stated various functions that are used in codes. "Void" means these functions do not spread any data. The end of the function name in brackets mean afferent types of variables. For example, 'void search (book*)' structure means search this function will receive a book data type, but do not spread any data.

For the "Main function" inside, the head node declares two link lists at fist. "Cardhead" is a pointer, which points to the card of the linked list structure. We set the first node that points to the list, put it in the "link" to be defined as "NULL". In the same way, then give

these node pointers to the allocated memory space (let them have the memory to store information). This time you'll find out a “User” pointer in the application of memory space, this “User” definition is a wide-changeable variable, it existed in the statement of ‘struct’. Next, I will illustrate the role of “User”. After all this are defined, the main function calls the "menu_welcome" function. This function can be found in many functions, it is a menu. Notice here, a very important detail is the inside of the parentheses variables. Here, “menu_welcomez” (cardhead) means to income “cardhead”, “head” points to their structure pointers.

In the “menu_welcome function”, first to display the menu, then use “switch” to connect different functions. For example, the user selects 1, then “menuz-welcome” this function will start the login function. In the “login” function, the first operation is to read the account information. Account information is the existence of a "card.txt" file. After reading the information, “card” chain on the current computer has all account information, when the user to enter a user name, every node in the linked list will compare user names and user input the user name for. (each node of the list is a structure, each structure contains the information of the user, for example, card->username is the user's user name) (StrCmp (A, B) function are used to compare, if A, B information are the same, it returns 0).If the user name exists, it will enter the next step, allows users to input the password, and then use the same “card->password “and password entered by the user to compare, if will show successfully login. Once the login is successful, it will give the address of node (i.e. pointer) is assigned to the pointer “User”. This is to say, after “User” landing, user information will be stored in this address. So if you want to visit the landing information or to modify the login information, it could be realized through the pointer to access the “User”.

When the login function is completed, the completion of will return to the “menu_welcome” function, and then you'll find out such code: "fflush (stdin)" means clear memory, there may exist some characters, will affect subsequent operation, so to make it empty. The "getchar", means that to get a character from the keyboard." System ("CLS") display means to clean everything on the screen. If there is no “getchar” instruction, the content of the screen will flash to pass, so need a “getchar” to control.

The last is "menu_normal", this menu is the user to login to complete the menu. There are a variety of functions that also with are performed by switch functions.

Register Function

–ask user to set user name and password

–choose the type of user

–open one file to store user information

–*by using loops aims to avoid two same user names*

Login Function

–ask user to enter their user name

–check the user name existed or not

–if the user name existed ask user to enter password, or not to check the user name again

–check the name or password are in pair to log in or not

–if the password was not in pair with the user name ask user to check password

–when the user successfully log in, open files to record user operation and information

use pointer to link the file that store account information

Add books Function

– ask the user to enter the ISBN and test the ISBN that has already been used or not.

– if the ISBN has not been used, ask user to enter book information. If the ISBN has already existed, users need to create a new ISBN.

—ask user to enter book information and store them

– open one text to record books information, at the same time, open the binary file to record the book information.

book information: ISBN, shelf, author, title,subject ,loan type, note, lend

Delete books function

– ask the user to enter the ISBN and test the ISBN that has already been existed or not.

– if the ISBN has not been existed, ask user to ensure book information. If the ISBN has already existed, show the book information and make sure users ensure to delete the book.

—choose the item and delete the book information

– open one text to record books information, at the same time, open the binary file to delete the book information.

Modify books information function

– ask the user to enter the ISBN and test the ISBN that has already been existed or not.

– if the ISBN has not been existed, ask user to ensure book information. If the ISBN has already existed, show the book information and make sure users ensure to choose the information needed to modify

– choose the item that needed to be modified and store them

– open one text to record books information, at the same time, open the binary file to modify the book information.

Use loops to read all information from the file into linked list

Delete user function

– ask the user to enter the user name and test the name that has already been existed or not.

– if the name has not been existed, ask user to ensure user information. If the name has already existed, make sure users ensure to delete the user.

– open one text to delete user information, at the same time, open the binary file to delete the user information

Search book function

*use the loop and link list, pointer to connect each search book information function
And combined in one menu. Search book information: ISBN, shelf, title,subject ,loan
Each book information form will link to the file that contains all books information,if the book already existed, it will display all information of the book*

– ask users to choose the item information to search books

– ask user t enter the book information to judge the existence of the book by linking to the data to defect

– If the book exists, it will give the information of the book. If the book does not exist, it

will tell the user the book does not exist.

Borrow book function

– ask users enter the ISBN and judge the book exists or not.

– If the book exists, it will give the information of the book and ensure the user to borrow or not. If the book does not exist, it will tell the user the book does not exist.

– if the book was borrowed, it would be recorded in history. At the same time, it will show the return information of the book

– open one text to record user book history information, at the same time, open the binary file to record the book information.

Return book function

*use link list and pointers to read book list information to check the book. Similarly, open account information file and store or modify account and book information *

– ask users enter the ISBN and judge the book exists or not.

– If the book exists, it will list the information about the book. If not exists, it will tell you to check book information

– the program will open the user account file to read history to judge whether the book was borrowed by user.

– if the history checked that the book was borrowed by users, it will ensure whether you want or not to return and notice your borrow information.

–if the history does not find the book tell the user that you did not borrow the book

– open one text to record user successful borrow book information, at the same time, open the binary file to record the book information.

View on loan book

–read the on loan book from files

–display the book list and information

View on loan book

–read the available book from files

–display the book list and information

4. Implementation:

See the C code in file sample.c with comments

C code

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

/*define struct of book*/
typedef struct book{
        char book[100];/* represents book name */
        char author[100];/* represents book author */
        char ISBN[100];/* represents book ISBN */
        char subject[100];/* represents book subject */
        char loan_type[100];/* represents book loan type */
        char shelf[100];/* represents the shefl that use to store books */
        int inventory;/* represents book inventory */
        int lend;/* represents book borrowed number */
        char Note[100];/* represents book notes */
        struct book *link;
} book;
/*define struct of user account*/
typedef struct card{
        char username[100];/* represent username */
        char password[100];/* represent password */
        char type;/* represent teacher or student */
        char history1[100];/* record the history of this account */
```

```
        struct card *link;
} card;/* define a data type which called "card" */

card *User;

void login(card*,book*);
void Register();
void menu_normal(card*,book*);
void menu_Admin(card*,book*);
void menu_welcome(card*,book*);
void AdminOption();
void My_information();
void search();
void Return(card*,book*);
void booklist_unborrowed(book*);
void booklist_borrowed(book*);
void add_book(book*);
void delete_book(book*);
void modify(book*);
void delete_user(card*);
void menu_searchbook();
void searchbook_name(book*);
void searchbook_author(book*);
void searchbook_ISBN(book*);
void searchbook_subject(book*);
void searchbook_loan(book*);
```

```
void searchbook_shelf(book*);

void borrow(card*,book*);


void login(card *cardhead,book *head){ /* this function make user to login the library system */

        int choice,a=0,b=0,c=1,d;

        char username[100],password[100],B;

        FILE *fp1;/* define a pointer pointing to a file */

        card *p1,*p2,*p3,*r1,*r2,*r3,*r4,*r5;/* define these pointer pointing to the card structure */

        r5=r1=r2=r3=r4=p1=p2=(card*)malloc(sizeof(card));/* allocate these addresses which pointers point to space to store information */

        fp1=fopen("card.txt","rb");/* open the file which include all the users' information */

        if(fp1==NULL)

        {

                printf("Thre is not any user in this system. Please register first!\n");

                exit(1);

        }

        p2=cardhead;/* make pointer "p2" point to the first node of linked list */

        while(!feof(fp1))/* use while loop to read all the information from file into linked list */

        {

                fread(p1,sizeof(card),1,fp1);

                p2->link=p1;

                p2=p1;

                p1=(card*)malloc(sizeof(card));

        }
```

```
        p2->link=NULL;
        fclose(fp1);/* close the file */
        while(b==0)/* b=0 is declared in order to do this loop at the first time */
        {
                p3=cardhead;/* make pointer "p3" point to the first node of linked list,
which has included all the users' information */
                printf("Please enter your username: \n");
                scanf("%s",&username);/* get the username */

                while(p3->link!=NULL) /* this loop aims to judge if this username exists
in the library system and if the password is correct */
                {
                        p3=p3->link;/* ensure every time pointer point to the next node
of linked list */
                        if(strcmp(p3->username,username)==0)/* use strcmp function to
judge if these two char string are the same */
                        {
                                b=1;/* if find this username, assign 1 to b (b=0 before) */
                                while(a!=1)/* this loop aims to judge if password is correct
(a=0 is declared before) */
                                {
                                        printf("Please enter your password: \n");
                                        scanf("%s",&password);
                                        if(strcmp(p3->password,password)==0)/* judge if
the password is correct */
                                        {
                                                printf("Log in successfully!\n");
```

```
                                        a=1;/* if the password is correct, assign 1
to a (a=0 before) */
                                        User=p3;/* transfer this address which
include this user's information to a global variable User, for convenience to use the
information of logged-in user */


                                    }
                                    else
                                    {
                                        printf("Password is not correct. Please enter
again\n");
                                    }
                                }
                            }
                        }
                    if(b==0)/* if the username does not exist, b will not be assigned value
and is still equal to 0 */
                        printf("This username does not exist! Please enter again.\n");
        }
}



void Register(card *cardhead) /* register a new account for users */
{
        int choice,a=0,b=0,c=1,d;
        char username[100],password[100];
        FILE *fp1;
```

```
    card *p1,*p2,*p3,*r1,*r2,*r3,*r4,*r5;

    r5=r1=r2=r3=r4=p1=p2=(card*)malloc(sizeof(book));

    fp1=fopen("card.txt","ab+"); /* the first time run this program, there is no file
"card.txt", in order to avoide error, create a file at the first time and will not cover the
previous file after*/

    if(fp1==NULL)

    {

        printf("Fail to open the file!");

    }

    fclose(fp1);

    fp1=fopen("card.txt","r"); /* open the file which include account information by
the type of read */

    if(fp1==NULL)

    {

        printf("Fail to open the file!\n");

        exit(1);

    }

    r1=cardhead; /* make the pointer "r1" point to the first node of the linked list */


    while(!feof(fp1))/* read the information from the file into linked list one node by
one node */

    {

        fread(r5,sizeof(card),1,fp1);

        r1->link=r5;

        r1=r5;

        r5=(card*)malloc(sizeof(card));

    }
```

```
        r1->link=NULL;/* make the pointer in the last node point to NULL */
        fclose(fp1);
        fp1=fopen("card.txt","ab");/* open this file again in order to add new
information into this file */
        if(fp1==NULL)
        {
                printf("Fail to open the file!");
        }
        r3=cardhead;/* make the pointer point to the first node of the linked list which
now has included all the accounts information */
        while(r3->link!=NULL)/* find the last node */
        {
                r3=r3->link;
        }
        while(c==1)/* this loop aims to avoid two same usernames */
        {
                r4=cardhead;
                printf("Please enter your username (Note: username cannot include
space): \n");
                scanf("%s",&username);
                while(r4->link!=NULL)/* check username one node by one node */
                {
                        r4=r4->link;
                        if(strcmp(r4->username,username)==0)/* if find the same
username */
                        {
                                printf("This username has existed. Please choose another
one.\n");
```

```
                        c=0;/* if find the same username, assign 0 to c (c=1
before) */
                    }
                }
                c++;/* if username is the same as existed username, this loop will
continue, or c will be equal to 2, so this loop will stop */
        }
        strcpy(r2->username,username);/* add information into the member of
structure */
        printf("Please enter your password: \n");
        fflush(stdin);
        gets(r2->password);
        printf("Are you the teacher or the student?\n");
        printf("1. Teacher\n");
        printf("2. Student\n");
        scanf("%d",&d);
        switch(d)
        {
        case 1:
                r2->type='T';/* if user choose teacher, the member "type" will be
assigned a character 'T' */
                break;
        case 2:
                r2->type='S';/* if user choose student, the member "type" will be
assigned a character 'S' */
                break;
        default:
                printf("Please choose correct item!\n");
```

```
        }
        r2->history1[0]='\0';/* empty the history of this account */
        fwrite(r2,sizeof(card),1,fp1);/* add the information of this new node into the file
*/
        fclose(fp1);
        r3->link=r2;
        r2->link=NULL;
        printf("Register successfully!\n");
}



void menu_normal(card *cardhead,book *head){ /* this menu is shown to logged-in
users */
        FILE *fp;
        book *p1,*p2;
        char bookname[100];
        int choice;
        p1=p2=(book*)malloc(sizeof(book));/* assign space for these address which
pointers point to */
        fp=fopen("librarybooks.txt","ab+");/* this step aims to avoid the error the first
time run this program when there is not file "librarybooks.txt", so use "ab+" type open
the file */
        if(fp==NULL)
        {
                printf("file is opened unseccessfully.\n");
                exit(1);
        }
```

```
        fclose(fp);
        fp=fopen("librarybooks.txt","rb");/* open the file which inlude all books
information in order to read them into linked list */
        if(fp==NULL)
        {
                printf("file is opened unseccessfully.\n");
                exit(1);
        }
        p2=head;/* make pointer "p2" point to the first node of linked list */
        while(!feof(fp))/* read information from the file into linked list one node by one
node */
        {
                fread(p1,sizeof(book),1,fp);
                p2->link=p1;
                p2=p1;
                p1=(book*)malloc(sizeof(book));
        }
        p2->link=NULL;
        fclose(fp);
       printf("          ************************************************\n");
        printf("          *                                  *\n");
        printf("          *          Please choose items:          *\n");
        printf("          *                                  *\n");
        printf("          *           1> My information          *\n");
        printf("          *           2> Search book            *\n");
        printf("          *           3> Borrow book            *\n");
```

```
        printf("        *        4> return book              *\n");
        printf("        *        5> View all available book  *\n");
        printf("        *        6> Log off                  *\n");
        printf("        *        7> Quit                     *\n");
        printf("        *                                    *\n");
       printf("        ************************************************\n");

        scanf("%d",&choice);
        switch(choice)/* through switch function to link other function */
        {
        case 1:
                system("cls");
                My_information();/* call the function which can display the personal
information */
                printf("Enter any key to return last menu.\n");
                fflush(stdin);
                getchar();/* get one key */
                system("cls");
                menu_normal(cardhead,head);/* after finishing the calling function, go
back the mune */
                break;
        case 2:
                system("cls");
                search(head);/* call the function which can search book */
                menu_normal(cardhead,head);
                break;
```

```
        case 3:
                system("cls");
                borrow(User,head);/* call the function which can borrow book */
                menu_normal(cardhead,head);
                break;
        case 4:
                system("cls");
                Return(User,head);/* call the function which can return book */
                menu_normal(cardhead,head);
                break;
        case 5:
                system("cls");
                booklist_unborrowed(head);/* call the function which can display the
book list of all available books */
                fflush(stdin);
                getchar();
                system("cls");
                menu_normal(cardhead,head);/* after finishing the calling function, go
back the mune */
                break;
        case 6:
                system("cls");
                User=NULL;/* initialize the global variable User ( clear the account
information ) */
                menu_welcome(cardhead,head);
                break;
        case 7:
```

```
            break;
        default:
            printf("Please choose correct option!\n");
        }
}

void menu_Admin(card *cardhead,book *head){ /* this menu is shown to administrator, which can add, delete books and manage users' accounts */
        int choice;
        printf("          ************************************************\n");
        printf("          *                                              *\n");
        printf("          *          Please choose items below:          *\n");
        printf("          *                                              *\n");
        printf("          *        1> Add books                          *\n");
        printf("          *        2> Delete books                       *\n");
        printf("          *        3> Modify informatons of books        *\n");
        printf("          *        4> Delete Users                       *\n");
        printf("          *        5> View all books on loan             *\n");
        printf("          *        6> Return to previous menu            *\n");
        printf("          *        7> Quit                               *\n");
        printf("          *                                              *\n");
        printf("          ************************************************\n");
        scanf("%d",&choice);
        switch(choice)/* user switch function to link other function */
        {
        case 1:
```

```
                system("cls");
                add_book(head);/* call the function which can add book */
                menu_Admin(cardhead,head);/* after finishing the calling function, go
back the mune */
                break;
        case 2:
                system("cls");
                delete_book(head);/* call the function which can delete book */
                printf("Please enter any key to continue...\n");
                fflush(stdin);
                getchar();
                system("cls");
                menu_Admin(cardhead,head);/* after finishing the calling function, go
back the mune */
                break;
        case 3:
                system("cls");
                modify(head);/* call the function which can modify the information of
books */
                printf("Please enter any key to continue...\n");
                fflush(stdin);
                getchar();
                system("cls");
                menu_Admin(cardhead,head);/* after finishing the calling function, go
back the mune */
                break;
        case 4:
```

```
                system("cls");
                delete_user(cardhead);/* call the function which can delete users'
accounts */
                printf("Enter any key to continue...\n");
                fflush(stdin);
                getchar();
                system("cls");
                menu_Admin(cardhead,head);/* after finishing the calling function, go
back the mune */
                break;
        case 5:
                system("cls");
                booklist_borrowed(head);/* call the function which can show all the book
on loan */
                printf("Enter any key to continue...\n");
                fflush(stdin);
                getchar();
                system("cls");
                menu_Admin(cardhead,head);/* after finishing the calling function, go
back the mune */
                break;
        case 6:
                system("cls");
                menu_welcome(cardhead,head);/* go back the first page */
                break;
        case 7:
                break;
```

```
        default:
                printf("Please choose correct option!");
        }
}

void menu_welcome(card *cardhead,book *head){ /* this menu is as the first page */
        int choice;
       printf("          ************************************************\n");
        printf("          *                                      *\n");
        printf("          *          Welcome to Library!         *\n");
        printf("          *                                      *\n");
        printf("          *        Please choose items below:    *\n");
        printf("          *                                      *\n");
        printf("          *            1> Log in                 *\n");
        printf("          *            2> Register               *\n");
        printf("          *         3> Administrator option      *\n");
        printf("          *            4> Quit                   *\n");
        printf("          *                                      *\n");
        printf("          *                    By group B8       *\n");
       printf("          ************************************************\n");
        scanf("%d",&choice);
        switch(choice) /* use switch function to link other function */
        {
        case 1:
                system("cls");
                login(cardhead,head);/* call the login function */
```

```
            fflush(stdin);
            getchar();
            system("cls");
            menu_normal(cardhead,head);/* after finishing login function means that
user login successfully, go to the logged-in menu */
            break;
        case 2:
            system("cls");
            Register(cardhead);/* call the function which can register a new account
*/
            printf("Enter any key to continue...\n");
            fflush(stdin);
            getchar();
            system("cls");
            menu_welcome(cardhead,head);/* after finishing the calling function, go
back the mune */
            break;
        case 3:
            system("cls");
            AdminOption(cardhead,head);/* call the function which can call the menu
for administrator */
            printf("Enter any key to continue...\n");
            getchar();
            system("cls");
            break;
        case 4:
            break;
```

```
        default:
                printf("Please choose correct items!\n");
        }
}
/*interface of search :search book by different choice */
void menu_searchbook()
{
        /*init searchbook panel*/
        printf("**************************************************************
***********\n");
        printf("*                    -SEARCH BOOK-                    *\n");
        printf("*                    1> Book Name                     *\n");
        printf("*                    2> Author                        *\n");
        printf("*                    3> ISBN                          *\n");
        printf("*                    4> Subject                       *\n");
        printf("*                    5> Loan type                     *\n");
        printf("*                    6> Shelf                         *\n");
        printf("*                    7> back                          *\n");
        printf("**************************************************************
***********\n");
}
```

```
main(){
        card *cardhead;
        book *head;
        head=(book*)malloc(sizeof(book));
        cardhead=(card*)malloc(sizeof(card));
        User=(card*)malloc(sizeof(card));
        cardhead->link=NULL;
        head->link=NULL;
        menu_welcome(cardhead,head);

}



void My_information(){ /* this function can displat the information of logged-in user */
        char teacher[]="teacher",student[]="student";
        printf("Username: %s\n",User->username);
        if(User->type=='S')/* judge the role of user by the type in the structure */
                printf("Type: %s\n",student);
        else
                printf("Type: %s\n",teacher);
        if(User->history1[0]=='\0')
                printf("No history.\n");
        else
        {
                printf("History: %s\n",User->history1);
```

```
                if(User->type=='S')
                        printf("      Time Limit: 30 days.");
                else
                        printf("      Time Limit: 60 days.");
        }



}

void search(book *head){/* declare defined funtion search() */
        int choose;/* for choose options */
        menu_searchbook();/* display menu of search book */
        printf(" Please choose a way to search book\n");
        scanf("%d",&choose);
        switch(choose)
        {
        case 1:
                system("cls");
                searchbook_name(head);/* search book by name */
                break;
        case 2:
                system("cls");
                searchbook_author(head);/* search book by author */
                break;
        case 3:
                system("cls");
```

```
                searchbook_ISBN(head);/* search book by ISBN */
                break;
        case 4:
                system("cls");
                searchbook_subject(head);/* search book by subject */
                break;
        case 5:
                system("cls");
                searchbook_loan(head);/* search book by loan */
                break;
        case 6:
                system("cls");
                searchbook_shelf(head);/* search book by shelf */
                break;
        case 7:
                system("cls");
                break;
        default:
                printf("Please choose correct option!\n");/* when enter choose wrong
option, display this sentence */
        }

}

void Return(card *cardhead, book *head)/* declare defined funtion search() */
{
```

```
FILE *fp,*fp1;/* fp represents libraryboos.txt ; fp1 represents card.txt */
int a,b;/* a for storing inventory ; b for storing lend number */
int b_choose;
char borrow_ISBN[100],tem[100];
int i=0;
book *p3;
card *c1,*c2,*c3,*c4;
p3=(book*)malloc(sizeof(book));
c4=c1=c2=c3=(card*)malloc(sizeof(card));
p3=head;
/*if book ptr is NULL it means no information for the book before */
if(head==NULL||head->link==NULL)
{
        printf("No books in the library!\n");
}
else
{
        /*return book by ISBN the book should be exist in the booklibrary*/
        printf("Please enter the book'ISBN which you want to return:\n");
        fflush(stdin);
        gets(borrow_ISBN);
        while(p3->link!=NULL)
        {
                p3=p3->link;
                /*if borrow ISBN exist in the book library*/
                if(strcmp(p3->ISBN,borrow_ISBN)==0)
```

```
{
        printf("find this book\n");
        printf("Book name: %s\n",p3->book);
        printf("Author name: %s\n",p3->author);
        printf("ISBN: %s\n",p3->ISBN);
        printf("Inventory: %d\n",p3->inventory);
        printf("Lend number: %d\n",p3->lend);
        printf("Note: %s\n",p3->Note);
        a=p3->inventory;
        b=p3->lend;
        /* if b>0 it means the book has been lent to this user */
        if(b>0)
        {
            printf("\nDo you want to return this book?\n");
            printf("1-Yes and 2-No\n");
            scanf("%d",&b_choose);
            getchar();
            /* the user will return this borrowed book*/
            if(b_choose==1)
            {

                if(strcmp(User->history1,p3->book)==0)
                {
                    /*open library and change the book
information*/
                    p3->inventory=a+1;
```

```
                                                    p3->lend=b-1;
                                                    strcpy(User->history1,"\0");
                                                    strcpy(tem,User->username);

                                                    fp1=fopen("card.txt","r");
                                                    if(fp1==NULL)
                                                    {
                                                            printf("Fail to open the
file!\n");

                                                            exit(1);
                                                    }
                                                    c1=cardhead;
                                                    while(!feof(fp1))
                                                    {
                                                            fread(c2,sizeof(card),1,fp1);
                                                            c1->link=c2;
                                                            c1=c2;

        c2=(card*)malloc(sizeof(card));

                                                    }
                                                    c1->link=NULL;
                                                    fclose(fp1);
                                                    c3=cardhead;
                                                    while(c3->link!=NULL)
                                                    {
                                                            c3=c3->link;
```

```
                    if(strcmp(c3->username,tem)==0)
                        strcpy(c3->history1,"\0");
                }
                /*open user and change the borrowed information*/
                fp1=fopen("card.txt","w");
                c4=cardhead;
                while(c4->link!=NULL)
                {
                    c4=c4->link;
                    fwrite(c4,sizeof(card),1,fp1);
                }
                fclose(fp1);
                printf("return book successfully!\n");
            }
            else
            {printf("You do not borrow this book!\n");
            menu_normal(cardhead,head);}


        }
        /*do not return*/
        if(b_choose==2)
        {
```

```
                                printf("cancel operation\n");
                                break;
                        }
                }
                /*the book did not lend to this user ever before*/
                if(b==0)
                {
                        printf("This book haven't borrowed!\n");
                }
                i=1;
        }
}
/*book not exist in the library*/
if(i==0)
{
        printf("this book is not in the library\n");
}
fp=fopen("librarybooks.txt","w");
if(fp==NULL)
{
        printf("Fail to open the file!\n");
        exit(1);
}
p3=head;
while(p3->link!=NULL)
{
```

```
                p3=p3->link;
                fwrite(p3,sizeof(book),1,fp);
            }
            fclose(fp);
            printf("Please enter any key to continue...");
            getchar();
            fflush(stdin);
            system("cls");
        }
}



void viewBook(book *head){}


void AdminOption(card *cardhead,book *head){
        menu_Admin(cardhead,head);
}


void add_book(book *head)/* declare a function add_book and use structure variable
book define prode head */
{
        int i=1;
        book *p1,*p2,*p3;/* declare struct book variable p1,p2,p3. For setting up a
linked list store data in librarybooks.txt */
        book *new,*last;/* declare struct book variable new and last. For setting up a
linked list output data in text file */
        FILE *fp;/* define a file pointer variable fp */
```

```
        fp=fopen("librarybooks.txt","ab+");/* "ab+" means for read and write open a
binary file */
        if(fp=NULL)
        {
                printf("Fail to open the file!\n");
                exit(1);
        }
        fclose(fp);
        p1=p2=p3=(book*)malloc(sizeof(book));/* Request a block of memory to p1,p2
and p3 of a give size for book */
        new=(book*)malloc(sizeof(book));/* Request a block of memory to new of a
give size for book */
        if(new==NULL)/* judge whether allocate memory */
        {
                printf("Out of memory\n");
                exit(1);
        }
        printf("Please enter the name of the book:\n");
        fflush(stdin);/* clear buffered stream */
        gets(new->book);
        printf("Please enter the author of the book:\n");
        fflush(stdin);
        gets(new->author);
        while(i==1)
        {
                printf("Please enter the ISBN of book:\n");
                fflush(stdin);
```

```
        gets(new->ISBN);
        fp=fopen("librarybooks.txt","rb");/* open a binary file namely librarybook
for read */
        if(fp==NULL)/* use if statement judge the file whether can open or not
*/
        {
                printf("file is opened unseccessfully.\n");
                exit(1);
        }
        p2=head;/* p2 point to head node */
        while(!feof(fp))/* before go to the end of librarybooks.txt */
        {
                fread(p1,sizeof(book),1,fp);/* use binary way enter a group of
data in p1 */
                p2->link=p1;/* give value of next node of p2 */
                p2=p1;/* let p2 point to p1 */
                p1=(book*)malloc(sizeof(book));/* Request a block of memory to
p1 of a give size for book */
        }
        p2->link=NULL;/* let last pointer equal to Null */
        fclose(fp);
        p3=head;/* let p3 point to head */
        if(head==NULL||head->link==NULL)/* no data in linked list */
        {
                printf("No books in the library!\n");
        }
        else
```

```
            {
                while(p3->link!=NULL)/* when output node p3, the value of next
node is NULL, out the loop */
                {
                    p3=p3->link;/* let p3 point to next node */
                    if(strcmp(p3->ISBN,new->ISBN)==0)/* if enter ISBN
eqult to existed ISBN */
                    {
                        printf("This ISBN already exists,Please change
another!\n");
                        i=0;
                    }
                }
            }
            i++;
        }
        printf("Please enter the subject of the book:\n");
        fflush(stdin);
        gets(new->subject);
        printf("Please enter the loan type(normal, short loan, no-take-out) of the
book:\n");
        fflush(stdin);
        gets(new->loan_type);
        printf("Please enter the shelf(A1~A10) of book:\n");
        fflush(stdin);
        gets(new->shelf);
        printf("Please enter the inventory(digit) of the book:\n");
```

```
        fflush(stdin);
        scanf("%d",&new->inventory);
        getchar();
        new->lend=0;/* when add a new book, the borrowed number is 0 */
        printf("Please enter the note (lost / damaged / ordered, etc.) of book:\n");
        fflush(stdin);
        gets(new->Note);
        fp=fopen("librarybooks.txt","ab");/* "ab" means add data at the end of binary
text */
        if(fp==NULL)
        {
                printf("file is opened unseccessfully.\n");
                exit(1);
        }
        fwrite(new,sizeof(book),1,fp);/* use binary way enter a group of data in file */
        fclose(fp);
        printf("Add book successfully\n");
        printf("Please enter any key to continue...");
        fflush(stdin);
        getchar();
        system("cls");/* system("cls") clears whatever is display on the screen */
}



void delete_book(book *head)/* declare a function delete_book and use structure
variable book define prode head */
```

```
{
        char bookISBN[100];
        FILE *fp;
        book *p1,*p2,*p3,*p4,*p5;
        int i=1;
        fp=fopen("librarybooks.txt","rb");/* open a binary file namely librarybook for
read */
        if(fp==NULL)
        {
                printf("There is no book in the system. Please add first!\n");
                exit(1);
        }
        p5=p4=p3=p1=p2=(book*)malloc(sizeof(book));
        p1=head;/* p1 point to head */
        while(!feof(fp))/* before go to the end of librarybooks.txt */
        {
                fread(p2,sizeof(book),1,fp);/* use binary way enter a group of data in p2
*/
                p1->link=p2;/* give value of next node of p1 */
                p1=p2;
                p2=(book*)malloc(sizeof(book));/* Request a block of memory to p2 of a
give size for book */
        }
        p1->link=NULL;/* let last pointer equal to Null */
        fclose(fp);
        while(i==1)
        {
```

```
            printf("Please enter the ISBN of book which will be deleted: \n");
            fflush(stdin);
            gets(bookISBN);
            p5=p4=head;/* create two linked list's list head */
            while(p4->link!=NULL)/* when output node p4, the value of next node is
NULL, out the loop */
            {
                  p4=p4->link;/* let p4 point to next node */
                  if(strcmp(bookISBN,p4->ISBN)==0)/* if enter ISBN eqult to
existed ISBN */
                  {
                        i=0;
                        p5->link=p4->link;/* p5 connect to p4 */
                        fp=fopen("librarybooks.txt","w");/* "w" means create a
new text */
                        if(fp==NULL)
                        {
                              printf("Fail to open the file!\n");
                              exit(1);
                        }
                        p3=head;/* create linked list p3 */
                        while(p3->link!=NULL)/* when output node p3, the value
of next node is NULL, out the loop */
                        {
                              p3=p3->link;/* let p3 point to next node */
                              fwrite(p3,sizeof(book),1,fp);/* use binary way
output a group of data in p3 */
                        }
```

```
                        fclose(fp);
                        printf("Delete book successfully!\n");
                }
                p5=p5->link;/* p5 point to next node */
        }
        if(i==1)
                printf("This book does not exist! Please enter anain!\n");

    }


}

void modify(book *head){ /* this function can modify all information of books */
    char modify_name[100];
    char modify_author[100];
    char modify_ISBN[100];
    char modify_subject[100];
    char modify_loan[100];
    char modify_shelf[100];
    int modify_inventory;
    int modify_lend;
    char modify_Note[100];
    char bookISBN[100];
    char modify_note[100];
    int i=0;
```

```
	int choose;
	FILE *fp;
	book *p1,*p2,*p3;
	p1=p2=p3=(book*)malloc(sizeof(book));
	fp=fopen("librarybooks.txt","rb");/* open the file which include books
information */
	if(fp==NULL)
	{
		printf("file is opened unseccessfully.\n");
		exit(1);
	}
	p2=head;
	while(!feof(fp)) /* this loop reads all information from the file into linked list */
	{
		fread(p1,sizeof(book),1,fp);
		p2->link=p1;
		p2=p1;
		p1=(book*)malloc(sizeof(book));
	}
	p2->link=NULL;
	fclose(fp);/* close the file */
	p3=head;
	if(head==NULL||head->link==NULL)
	{
		printf("No books in the library!\n");
	}
```

```
        else
        {
                printf("Please enter the book'ISBN which you want to modify:\n");
                fflush(stdin);
                gets(bookISBN);
                while(p3->link!=NULL)/* this loop aims to make the pointer point to the
next node one by one */
                {
                        p3=p3->link;
                        if(strcmp(bookISBN,p3->ISBN)==0) /* judge if the current node
is the target */
                        {
                                printf("find this book\n"); /* show the information of this
book */
                                printf("Book name: %s\n",p3->book);
                                printf("Author name: %s\n",p3->author);
                                printf("ISBN: %s\n",p3->ISBN);
                                printf("Subject: %s\n",p3->subject);
                                printf("Loan type: %s\n",p3->loan_type);
                                printf("Shelf: %s\n",p3->shelf);
                                printf("Inventoryt: %d\n",p3->inventory);
                                printf("Lend number: %d\n",p3->lend);
                                printf("Note: %s\n",p3->Note);
                                do /* use do-while loop to do this opetration repeatedly */
                                {
                                        printf("\nDo you want to modify which
information?\n");
```

```
                printf("****************************\n");
                printf("*      1>book name         *\n");
                printf("*      2>book author       *\n");
                printf("*      3>book ISBN         *\n");
                printf("*      4>book subject      *\n");
                printf("*      5>book loan type    *\n");
                printf("*      6>book shelf        *\n");
                printf("*      7>book inventory    *\n");
                printf("*      8>book lend         *\n");
                printf("*      9>book note         *\n");
                printf("*                          *\n");
                printf("*      10>MODIFY           *\n");
                printf("****************************\n");
                fflush(stdin);
                scanf("%d",&choose);
                getchar();
                switch(choose)/* use switch function to link other
function */
                {
                case 1:
                        printf("origin book name: %s\n",p3->book);
                        printf("Please enter new book name\n");
                        strcpy(p3->book,"");
                        gets(modify_name);/* get the book name
and store it into char shring "modify_name" */
                        strcpy(p3->book,modify_name);/* copy this
name into member "book" of the current node */
```

```
                    break;
                case 2:
                    printf("origin book author: %s\n",p3->author);
                    printf("Please enter new book author\n");
                    strcpy(p3->author,"");
                    gets(modify_author);
                    strcpy(p3->author,modify_author);/* copy this author name into member "author" of the current node */
                    break;
                case 3:
                    printf("origin book ISBN: %s\n",p3->ISBN);
                    printf("Please enter new book ISBN\n");
                    strcpy(p3->ISBN,"");
                    gets(modify_ISBN);
                    strcpy(p3->ISBN,modify_ISBN);/* copy this new ISBN into member "ISBN" of the current node */
                    break;
                case 4:
                    printf("origin book subject: %s\n",p3->subject);
                    printf("Please enter new book subject\n");
                    strcpy(p3->subject,"");
                    gets(modify_subject);
                    strcpy(p3->subject,modify_subject);/* copy this new subject into member "subject" of the current node */
                    break;
                case 5:
```

```
                printf("origin book loan type: %s\n",p3->loan_type);
                printf("Please enter new book loan type\n");
                strcpy(p3->loan_type,"");
                gets(modify_loan);
                strcpy(p3->loan_type,modify_loan);/* copy this new type into member "loan_type" of the current node */
                break;
            case 6:
                printf("origin book shelf: %s\n",p3->shelf);
                printf("Please enter new book shelf\n");
                strcpy(p3->shelf,"");
                gets(modify_shelf);
                strcpy(p3->shelf,modify_shelf);/* copy this new shelf into member "shelf" of the current node */
                break;
            case 7:
                printf("origin book inventory: %d\n",p3->inventory);
                printf("Please enter new book inventory\n");
                scanf("%d",&modify_inventory);
                getchar();
                p3->inventory=modify_inventory;/* assign the new inventory to the member "inventory" of the current node */
                break;
            case 8:
                printf("origin book lend number: %d\n",p3->lend);
```

```
                                        printf("Please enter new book lend
number\n");
                                        scanf("%d",&modify_lend);
                                        getchar();
                                        p3->lend=modify_lend;/* assign the new
lend to the member "lend" of the current node */
                                        break;
                                case 9:
                                        printf("origin book note: %s\n",p3->Note);
                                        printf("Please enter new book note\n");
                                        strcpy(p3->Note,"");
                                        gets(modify_note);
                                        strcpy(p3->Note,modify_note);/* copy this
new note into member "Note" of the current node */
                                        break;
                                case 10:
                                        printf("Modify successfully\n");
                                        break;
                                default:
                                        printf("Please enter the correct integer
number.");
                                }
                        }
                        while(choose!=10);/* this is the judge condition of the do-
while loop */
                        i=1;/* if this do-while loop completed, which means the
information has been saved into the linked list, assign 1 to i (i=0 before) */
                }
```

```
                }
                if(i==0)/* if the ISBN is not correct, do-while loop will not be executed,
so the value of i is still equal to 0 */
                {
                        printf("this book is not in the library\n");
                }
                if(i==1)/* if the information has been saved into linked list, then, write
the new information of the linked list into file and cover the previous data */
                {
                        fp=fopen("librarybooks.txt","w");
                        if(fp==NULL)
                        {
                                printf("Fail to open the file!\n");
                                exit(1);
                        }
                        p3=head;
                        while(p3->link!=NULL) /* this loop writes the information into the
file */
                        {
                                p3=p3->link;
                                fwrite(p3,sizeof(book),1,fp);
                        }
                        fclose(fp);

                }
        }
}
```

```
void delete_user(card* cardhead){ /* this function aims to delete the accounts
information */
        char username[100];
        FILE *fp;
        card *p1,*p2,*p3,*p4,*p5;
        int i=1;
        User=NULL;
        fp=fopen("card.txt","rb");/* open the file wihch include all accounts information
*/
        if(fp==NULL)
        {
                printf("Fail to open the file!\n");
                exit(1);
        }
        p5=p4=p3=p1=p2=(card*)malloc(sizeof(card));
        p1=cardhead;/* make the pointer "p1" point to the first node of the linked list */
        while(!feof(fp))/* read information from file into linked list one node by one node
*/
        {
                fread(p2,sizeof(card),1,fp);
                p1->link=p2;
                p1=p2;
                p2=(card*)malloc(sizeof(card));
        }
        p1->link=NULL;
        fclose(fp);
```

```
        while(i==1)/* this loop aims to avoid that administrator enter a wrong username
*/
        {
                printf("Please enter the username which will be deleted: \n");
                scanf("%s",&username);
                p5=p4=cardhead;

                while(p4->link!=NULL)
                {
                        p4=p4->link;
                        if(strcmp(username,p4->username)==0) /* if find this username
*/
                        {
                                i=0;/* assign 0 to a (a=1 before) */
                                p5->link=p4->link;/* delete the node which cotain the
information fo this username */
                                fp=fopen("card.txt","w");/* use write type to open the file
in order to cover the previous data */
                                if(fp==NULL)
                                {
                                        printf("Fail to open the file!\n");
                                        exit(1);
                                }
                                p3=cardhead;
                                while(p3->link!=NULL)/* write the linked list which has
deleted the user account into the file again */
                                {
                                        p3=p3->link;
```

```
                                fwrite(p3,sizeof(card),1,fp);
                        }
                        fclose(fp);
                        printf("Delete user successfully!\n");
                }
                p5=p5->link;/* this step aims to make p5 always lags p4 one node before if judge */
        }

        if(i==1)
                printf("This username does not exist! Please enter again!\n");/* if there is not this username, the value of i will not change */

    }

}

void searchbook_name(book *head){/* declare a function searchbook_name and use structure variable book define prode head */
    char bookname[100];
    book *p3;
    int i=0;
    p3=(book*)malloc(sizeof(book));
    p3=head;/* p3 point to head */
    if(head==NULL||head->link==NULL)/* no data in linked list */
    {
        printf("No books in the library!\n");
```

```
            exit(1);
        }
        else
        {
            printf("Please enter the name of book:\n");
            fflush(stdin);/* clear buffered stream */
            gets(bookname);
            while(p3->link!=NULL)/* when output node p3, the value of next node is
NULL, out the loop */
            {
                p3=p3->link;/* let p3 point to next node */
                if(strcmp(p3->book,bookname)==0)/* if enter book name eqult
to existed book name */
                {
                    printf("find following books\n");
                    printf("Book name: %s\n",p3->book);
                    printf("Author name: %s\n",p3->author);
                    printf("ISBN: %s\n",p3->ISBN);
                    printf("Subject: %s\n",p3->subject);
                    printf("Loan type: %s\n",p3->loan_type);
                    printf("Shelf: %s\n",p3->shelf);
                    printf("Inventory: %d\n",p3->inventory);
                    printf("Lend number: %d\n",p3->lend);
                    printf("Note: %s\n",p3->Note);
                    i=1;/* this book is existed */
                }
```

```
                }
                if(i==0)
                {
                        printf("this book is not in the library\n");
                }
                printf("Please enter any key to continue...");
                getchar();
                fflush(stdin);
                system("cls");/* system("cls") clears whatever is display on the screen */
        }
}


void searchbook_author(book* head)/* declare a function searchbook_author and use
structure variable book define prode head */
{
        char bookauthor[100];
        int i=0;
        book* p3;/* p3 point to head */
        p3=(book*)malloc(sizeof(book));
        p3=head;
        if(head==NULL||head->link==NULL)/* no data in linked list */
        {
                printf("No books in the library!\n");
        }
        else
```

```
        {
                printf("Please enter the author of book:\n");
                fflush(stdin);/* clear buffered stream */
                gets(bookauthor);
                while(p3->link!=NULL)/* when output node p3, the value of next node is
NULL, out the loop */
                {
                        p3=p3->link;/* let p3 point to next node */
                        if(strcmp(p3->author,bookauthor)==0) /* if enter book author
eqult to existed book author */
                        {
                                printf("find following books\n");
                                printf("Book name: %s\n",p3->book);
                                printf("Author name: %s\n",p3->author);
                                printf("ISBN: %s\n",p3->ISBN);
                                printf("Subject: %s\n",p3->subject);
                                printf("Loan type: %s\n",p3->loan_type);
                                printf("Shelf: %s\n",p3->shelf);
                                printf("Inventory: %d\n",p3->inventory);
                                printf("Lend number: %d\n",p3->lend);
                                printf("Note: %s\n",p3->Note);
                                i=1;/* this book is existed */
                        }
                }
                if(i==0)
                {
```

```
                printf("this book is not in the library\n");
        }
        printf("Please enter any key to continue...");
        getchar();
        fflush(stdin);
        system("cls"); /* system("cls") clears whatever is display on the screen */
    }
}

void searchbook_ISBN(book *head)/* declare a function searchbook_ISBN and use
structure variable book define prode head */
{
    char bookISBN[100];
    int i=0;
    book *p3;
    p3=(book*)malloc(sizeof(book));
    p3=head;/* p3 point to head */
    if(head==NULL||head->link==NULL)/* no data in linked list */
    {
        printf("No books in the library!\n");
    }
    else
    {
        printf("Please enter the ISBN of book:\n");
        fflush(stdin);/* clear buffered stream */
        gets(bookISBN);
```

```
        while(p3->link!=NULL)/* when output node p3, the value of next node is
NULL, out the loop */
        {
                p3=p3->link;/* let p3 point to next node */
                if(strcmp(p3->ISBN,bookISBN)==0) /* if enter book ISBN eqult to
existed book ISBN */
                {
                        printf("find following books\n");
                        printf("Book name: %s\n",p3->book);
                        printf("Author name: %s\n",p3->author);
                        printf("ISBN: %s\n",p3->ISBN);
                        printf("Subject: %s\n",p3->subject);
                        printf("Loan type: %s\n",p3->loan_type);
                        printf("Shelf: %s\n",p3->shelf);
                        printf("Inventory: %d\n",p3->inventory);
                        printf("Lend number: %d\n",p3->lend);
                        printf("Note: %s\n",p3->Note);
                        i=1;/* this book is existed */
                }
        }
        if(i==0)
        {
                printf("this book is not in the library\n");
        }
        printf("Please enter any key to continue...");
        getchar();
```

```
                fflush(stdin);
                system("cls"); /* system("cls") clears whatever is display on the screen */
        }
}



void searchbook_subject(book* head)/* declare a function searchbook_subject and use structure variable book define prode head */
{
        char booksubject[100];
        int i=0;
        book *p3;
        p3=(book*)malloc(sizeof(book));
        p3=head;/* p3 point to head */
        if(head==NULL||head->link==NULL)/* no data in linked list */
        {
                printf("No books in the library!\n");
        }
        else
        {
                printf("Please enter the subject of book:\n");
                fflush(stdin);/* clear buffered stream */
                gets(booksubject);
                while(p3->link!=NULL)/* when output node p3, the value of next node is NULL, out the loop */
                {
```

```
                p3=p3->link;/* let p3 point to next node */
                if(strcmp(p3->subject,booksubject)==0) /* if enter book subject eqult to existed book subject */
                {
                        printf("find following books\n");
                        printf("Book name: %s\n",p3->book);
                        printf("Author name: %s\n",p3->author);
                        printf("ISBN: %s\n",p3->ISBN);
                        printf("Subject: %s\n",p3->subject);
                        printf("Loan type: %s\n",p3->loan_type);
                        printf("Shelf: %s\n",p3->shelf);
                        printf("Inventory: %d\n",p3->inventory);
                        printf("Lend number: %d\n",p3->lend);
                        printf("Note: %s\n",p3->Note);
                        i=1;/* this book is existed */
                }
        }
        if(i==0)
        {
                printf("this book is not in the library\n");
        }
        printf("Please enter any key to continue...");
        getchar();
        fflush(stdin);
        system("cls"); /* system("cls") clears whatever is display on the screen */
}
```

```
}


void searchbook_loan(book *head)/* declare a function searchbook_loan and use
structure variable book define prode head */
{
        char bookloan[100];
        int i=0;
        book *p3;
        p3=(book*)malloc(sizeof(book));
        p3=head;/* p3 point to head */
        if(head==NULL||head->link==NULL)/* no data in linked list */
        {
                printf("No books in the library!\n");
        }
        else
        {
                printf("Please enter the loan type of book:\n");
                fflush(stdin);/* clear buffered stream */
                gets(bookloan);
                while(p3->link!=NULL)/* when output node p3, the value of next node is
NULL, out the loop */
                {
                        p3=p3->link;/* let p3 point to next node */
                        if(strcmp(p3->loan_type,bookloan)==0)  /* if enter book loan
type eqult to existed book loan type */
                        {
                                printf("find following books\n");
```

```
                        printf("Book name: %s\n",p3->book);
                        printf("Author name: %s\n",p3->author);
                        printf("ISBN: %s\n",p3->ISBN);
                        printf("Subject: %s\n",p3->subject);
                        printf("Loan type: %s\n",p3->loan_type);
                        printf("Shelf: %s\n",p3->shelf);
                        printf("Inventory: %d\n",p3->inventory);
                        printf("Lend number: %d\n",p3->lend);
                        printf("Note: %s\n",p3->Note);
                        i=1;  /* this book is existed */
                }
        }
        if(i==0)
        {
                printf("this book is not in the library\n");
        }
        printf("Please enter any key to continue...");
        getchar();
        fflush(stdin);
        system("cls");/* system("cls") clears whatever is display on the screen */
    }
}


void searchbook_shelf(book *head)/* declare a function searchbook_shelf and use
structure variable book define prode head */
{
```

```
	char bookshelf[100];
	int i=0;
	book *p3;
	p3=(book*)malloc(sizeof(book));
	p3=head;/* p3 point to head */
	if(head==NULL||head->link==NULL)/* no data in linked list */
	{
		printf("No books in the library!\n");
	}
	else
	{
		printf("Please enter the book shelf of book:\n");
		fflush(stdin);/* clear buffered stream */
		gets(bookshelf);
		while(p3->link!=NULL)/* when output node p3, the value of next node is
NULL, out the loop */
		{
			p3=p3->link;/* let p3 point to next node */
			if(strcmp(p3->shelf,bookshelf)==0)/* if enter book shelf eqult to
existed book shelf */
			{
				printf("find following books\n");
				printf("Book name: %s\n",p3->book);
				printf("Author name: %s\n",p3->author);
				printf("ISBN: %s\n",p3->ISBN);
				printf("Subject: %s\n",p3->subject);
```

```
                        printf("Loan type: %s\n",p3->loan_type);
                        printf("Shelf: %s\n",p3->shelf);
                        printf("Inventory: %d\n",p3->inventory);
                        printf("Lend number: %d\n",p3->lend);
                        printf("Note: %s\n",p3->Note);
                        i=1;   /* this book is existed */
                    }
                }
                if(i==0)
                {
                    printf("this book is not in the library\n");
                }
                printf("Please enter any key to continue...");
                getchar();
                fflush(stdin);
                system("cls"); /* system("cls") clears whatever is display on the screen */
        }
}

void borrow(card* cardhead, book *head) /* borrow function */
{
        FILE *fp,*fp1;
        int a,b;
        int b_choose;
        char borrow_ISBN[100],tem[100];
        int i=0;
```

```
        book *p3;
        card *c1,*c2,*c3,*c4,*c5;
        c5=c4=c3=c1=c2=(card*)malloc(sizeof(card));
        p3=(book*)malloc(sizeof(book));
        p3=head;
        if(head==NULL||head->link==NULL)
        {
                printf("No books in the library!\n");
        }
        else /* user can borrow a book by entering its ISBN */
        {
                printf("Please enter the book'ISBN which you want to borrow:\n");
                fflush(stdin);
                gets(borrow_ISBN);
                while(p3->link!=NULL) /* move the pointer point to the next node one
time by one time */
                {
                        p3=p3->link;
                        if(strcmp(p3->ISBN,borrow_ISBN)==0)   /* judge if the current
node store the target book information */
                        {
                                printf("find this book\n");       /* if find this node, display
all information of this node */
                                printf("Book name: %s\n",p3->book);
                                printf("Author name: %s\n",p3->author);
                                printf("ISBN: %s\n",p3->ISBN);
                                printf("Inventory: %d\n",p3->inventory);
```

```
                printf("Lend number: %d\n",p3->lend);
                printf("Note: %s\n",p3->Note);
                a=p3->inventory;/* assign the inventory of the current
node to variable "a" */
                b=p3->lend;/* assign the lend of the current node to
variable "b" */
                if(a>0)/* judge if the inventory of this book more than
zero */
                {
                        printf("\nDo you want to borrow this book?\n");
                        printf("1-Yes and 2-No\n");
                        scanf("%d",&b_choose);
                        getchar();
                        if(b_choose==1)/* if user want to borrow this book
*/
                        {
                                if(User->history1[0]=='\0')/* judge if this
user's history is empty (One account can keep one book per time) */
                                {
                                        p3->inventory=a-1;/* decrease the
inventory by 1*/
                                        p3->lend=b+1;/* increase the lend
by 1*/
                                        strcpy(User->history1,p3->book);/*
copy this book name to the account's history (this step aims to user can see this history
in his information after borrowing book */
                                        strcpy(tem,User->username);/*
copy the current user's username to a variable "tem" in order to find this user in the
following operation */
```

```
                                        fp1=fopen("card.txt","r");/* open
the account file and read information from file into linked list */
                                        if(fp1==NULL)
                                        {
                                                printf("Fail to open the
file!\n");
                                                exit(1);
                                        }
                                        c1=cardhead;
                                        while(!feof(fp1))/* read the
information from the file into linked list */
                                        {
                                                fread(c2,sizeof(card),1,fp1);
                                                c1->link=c2;
                                                c1=c2;

        c2=(card*)malloc(sizeof(card));
                                        }
                                        c1->link=NULL;
                                        fclose(fp1);
                                        c3=cardhead;
                                        while(c3->link!=NULL) /* find the
user who borrow this book */
                                        {
                                                c3=c3->link;
                                                if(strcmp(c3-
>username,tem)==0)/* judge if the user is the user who borrow this book */
                                                {
```

```
                                                    strcpy(c3-
>history1,p3->book);/* if this user is the user who borrow this book, copy the book
name into the member "history1" of this node */
                                                        }
                                                    }
                                                    fp1=fopen("card.txt","w");/* open
the file again and cover the previous data */
                                                    if(fp1==NULL)
                                                    {
                                                        printf("Fail to open the
file!\n");
                                                        exit(1);
                                                    }
                                                    c4=cardhead;
                                                    while(c4->link!=NULL)/* write new
information which contain the history into the file */
                                                    {
                                                        c4=c4->link;
                                                        fwrite(c4,sizeof(card),1,fp1);
                                                    }


                        fclose(fp1);
                                                   printf("borrow book successfully!\n");
                                                    if(User->type=='T')/* judge the role
of this user */
                                                    {
                                                        printf("You are a teacher and
you can keep this book for 60 days. Please return it on time.\n");
```

```
                                        }
                                        else
                                                printf("You are a student and
you can keep this book for 30 days. Please return it on time.\n");
                                }
                                else
                                {
                                        printf("You can only borrow one
book per time! Please return your book first!\n");/* if the user's history is not empty,
which means he has kept a book,
                                        he will not be allowed to borrow
another one */
                                        getchar();
                                        system("cls");
                                        menu_normal(cardhead,head);/* go
back the menu */
                                }
                        }
                        if(b_choose==2) /* if user does not want to
borrow book */
                        {
                                printf("cancel operation\n");
                                break;
                        }
                }
                else if(a==0)
                {
```

```
                                        printf("Sorry, this book borrowed out\n");/* if the
inventory of this book is zero, tell the user this book is borrowed out */
                                }
                                i=1; /* if find this book, assign 1 to i (i=0 before) */
                        }
                }
                if(i==0) /* which means this book is not found in the library */
                {
                        printf("this book is not in the library\n");
                }
                fp=fopen("librarybooks.txt","w");/* open the file to cover the previous
data */
                if(fp==NULL)
                {
                        printf("Fail to open the file!\n");
                        exit(1);
                }
                p3=head;
                while(p3->link!=NULL) /* write the new book information to the file */
                {
                        p3=p3->link;
                        fwrite(p3,sizeof(book),1,fp);
                }
                fclose(fp);
                printf("Please enter any key to continue...");
                getchar();
```

```
            fflush(stdin);
            system("cls");
        }
}
```

```
void booklist_unborrowed(book *head)/* declare a function booklist_unborrowed and use structure variable book define prode head */
{
        int a,i=0;/* a for store the book inventory */
        book *p3;
        p3=(book*)malloc(sizeof(book));
        p3=head;/* p3 point to head */
        if(head==NULL||head->link==NULL)/* no data in linked list */
        {
            printf("No books in the library!\n");
        }
        else
        {
            while(p3->link!=NULL)/* when output node p3, the value of next node is NULL, out the loop */
            {
                p3=p3->link;/* let p3 point to next node */
                if(a=p3->inventory>0)  /* book inventory bigger than 0, it means these books are unborrowed */
                {
                    i=1;/* find unborrowed book */
```

```
                        printf("find unborrowed book\n");
                        printf("Book name: %s\n",p3->book);
                        printf("ISBN: %s\n",p3->ISBN);
                        printf("Inventoryt: %d\n",p3->inventory);
                        printf("Lend number: %d\n\n",p3->lend);
                }
        }
        if(i==0)
        {
                printf("There is not any book left.\n");
        }
    }
}

void booklist_borrowed(book *head)/* declare a function booklist_borrowed and use structure variable book define prode head */
{
    int b,i=0;/* b for store the book inventory */
    FILE *fp;
    book *p1,*p2,*p3;
    p1=p2=p3=(book*)malloc(sizeof(book));
    fp=fopen("librarybooks.txt","rb"); /* open a binary file namely librarybook for read */
    if(fp==NULL)
    {
        printf("file is opened unseccessfully.\n");
```

```
            exit(1);
        }
        p2=head;
        while(!feof(fp))
        {
            fread(p1,sizeof(book),1,fp);
            p2->link=p1;
            p2=p1;
            p1=(book*)malloc(sizeof(book));
        }
        p2->link=NULL;
        fclose(fp);
        p3=head;/* open a binary file namely librarybook for read */
        if(head==NULL||head->link==NULL)/* no data in linked list */
        {
            printf("No books in the library!\n");
        }
        else
        {
            while(p3->link!=NULL)/* when output node p3, the value of next node is
NULL, out the loop */
            {
                p3=p3->link;/* let p3 point to next node */
                if(b=p3->lend>0)  /* book lend number bigger than 0, it means
these books are borrowed */
                {
```

```
                        i=1;/* find borrowed book */
                        printf("find borrowed book\n");
                        printf("Book name: %s\n",p3->book);
                        printf("ISBN: %s\n",p3->ISBN);
                        printf("Inventoryt: %d\n",p3->inventory);
                        printf("Lend number: %d\n",p3->lend);
                }
        }
        if(i==0)
        {
                printf("There is not any book on loan.\n");
        }
    }
}
```

5. Testing:

The C program was tested by carrying out a set of experiments; and the C program output was verified successfully. For instance,

Main menu:

When open the program, main menu will display on the screen.

Register function:

```
Please enter your username (Note: username cannot include space):
student_user
Please enter your password:
xjtlu123
Are you the teacher or the student?
1. Teacher
2. Student
2
Register successfully!
Enter any key to continue...
```

Student account

```
Please enter your username (Note: username cannot include space):
teacher_user
Please enter your password:
xjtlu123
Are you the teacher or the student?
1. Teacher
2. Student
1
Register successfully!
Enter any key to continue...
```

Teacher account

1- Enter user name.
2- Enter password.
3- Register teacher or student account. Enter 1 for teacher and 2 for student.

```
Please enter your username (Note: username cannot include space):
teacher_user
This username has existed. Please choose another one.
Please enter your username (Note: username cannot include space):
```

When register existed account, program will give notice to user to change the user name.

Log in function:

Now, there are two registered accounts in library system: student_user xjtlu123 and teacher_user xjtlu123

```
Please enter your username:
student_user
Please enter your password:
xjtlu123
Log in successfully!
```

Log in student_user account

```
Please enter your username:
teacher_user
Please enter your password:
xjtlu123
Log in successfully!
```

Log in teacher_user account

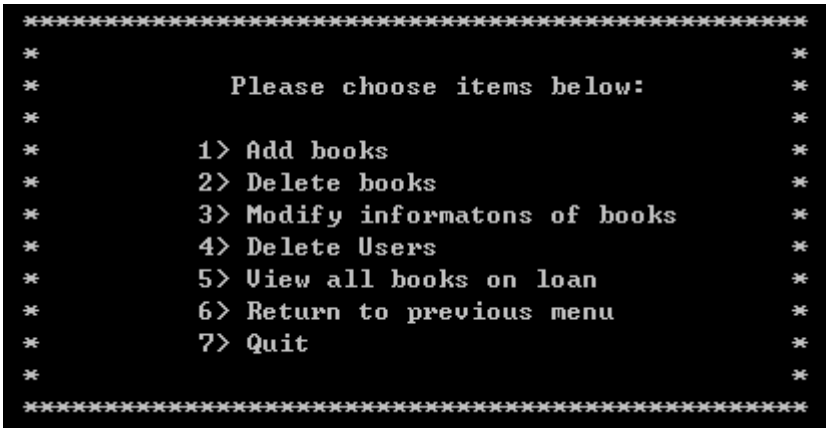
```
Please enter your username:
0000
This username does not exist! Please enter again.
Please enter your username:
```

When enter the account which not exited in library system

```
Please enter your username:
teacher_user
Please enter your password:
0000
Password is not correct. Please enter again
Please enter your password:
```

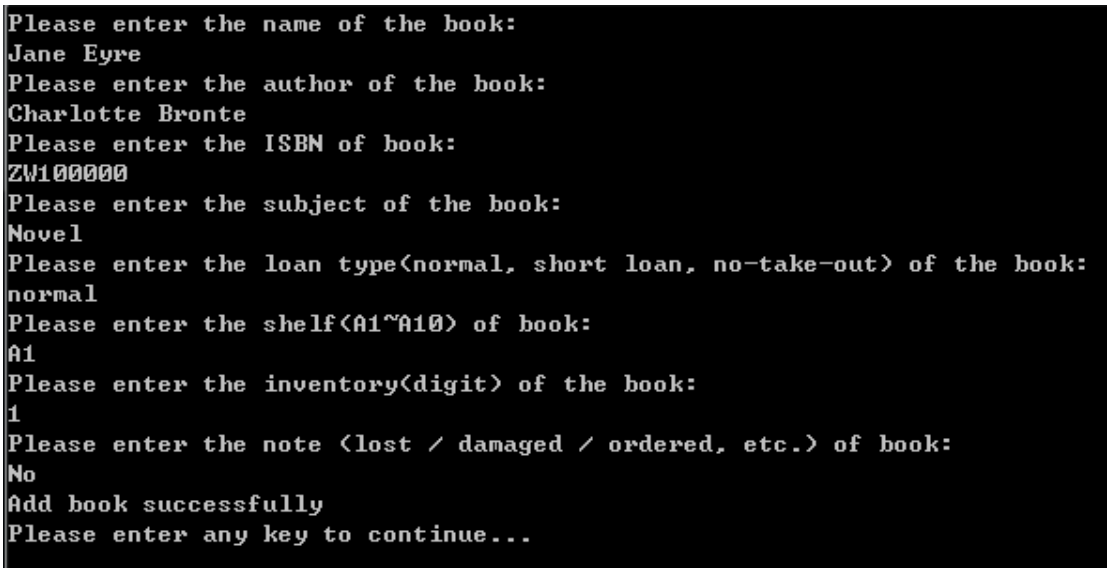
When enter wrong password

Administrator option menu:



When choose Administrator option, this administrator will display on screen.

Add books function:



Add book << Jane Eyre>>

```
Please enter the name of the book:
The C Programming Language
Please enter the author of the book:
Brian W. Kernighan & Dennis M. Ritchie
Please enter the ISBN of book:
ZW100001
Please enter the subject of the book:
EEE101
Please enter the loan type(normal, short loan, no-take-out) of the book:
normal
Please enter the shelf(A1~A10) of book:
A2
Please enter the inventory(digit) of the book:
2
Please enter the note (lost / damaged / ordered, etc.) of book:
No
Add book successfully
Please enter any key to continue...
```

Add book <<The C Programming Language>>

1- Enter the book name
2- Enter the author
3- Enter the ISBN
4- Enter the subject
5- Enter the loan type
6- Enter the shelf area
7- Enter the inventory of the book(enter number)
8- Enter notes of the book

```
Please enter the name of the book:
test
Please enter the author of the book:
r
Please enter the ISBN of book:
ZW100001
This ISBN already exists,Please change another!
Please enter the ISBN of book:
```

When enter existed ISBN, program will give notice to user to change another one.

Delete books function:

For testing this function, add a book at the beginning and search it, make sure this book add in the library, then use delete books function, finally, search it again, confirm this book was deleted.

```
Please enter the ISBN of book:
ZW100003
find following books
Book name: for delete
Author name: r
ISBN: ZW100003
Subject: EEE101
Loan type: short loan
Shelf: A3
Inventoryt: 1
Lend number: 0
Note: damaged
Please enter any key to continue...
```

Add delete book

```
Please enter the ISBN of book which will be deleted:
ZW100003
Delete book successfully!
Please enter any key to continue...
```

Use Delete books function

```
Please enter the ISBN of book:
ZW100003
this book is not in the library
Please enter any key to continue...
```

Delete successful

Modify information of books function:

For testing this function, add a book at the beginning and search it, record the details about this book, then use modify information of books function modify each information , finally, search it again, confirm the detail about this book was modified.

```
Please enter the ISBN of book:
SW100004
find following books
Book name: for modify
Author name: r
ISBN: SW100004
Subject: eee
Loan type: no-take-out
Shelf: A4
Inventoryt: 10
Lend number: 0
Note: No
Please enter any key to continue...
```

Original book

```
Please enter the book'ISBN which you want to modify:
SW100004
find this book
Book name: for modify
Author name: r
ISBN: SW100004
Subject: eee
Loan type: no-take-out
Shelf: A4
Inventoryt: 10
Lend number: 0
Note: No

Do you want to modify which information?
******************************
*      1>book name           *
*      2>book author         *
*      3>book ISBN           *
*      4>book subject        *
*      5>book loan type      *
*      6>book shelf          *
*      7>book inventory      *
*      8>book lend           *
*      9>book note           *
*                            *
*      10>MODIFY             *
******************************
```

Use modify function

```
1
origin book name: for modify
Please enter new book name
Gone with the wind
```

Modify book name

```
2
origin book author: r
Please enter new book author
Margaret Mitchell
```

Modify book author

```
3
origin book ISBN: SW100004
Please enter new book ISBN
ZW100004
```

Modify book ISBN

```
4
origin book subject: eee
Please enter new book subject
Novel
```

Modify book subject

```
5
origin book loan type: no-take-out
Please enter new book loan type
normal
```

Modify book loan type

```
6
origin book shelf: A4
Please enter new book shelf
A1
```

Modify book shelf

```
7
origin book inventory: 10
Please enter new book inventory
3
```

Modify book inventory

```
9
origin book note: No
Please enter new book note
Modify old book to new book
```

Modify book note

```
10
Modify successfully
Please enter any key to continue...
```

Choose MODIFY to finish modify part

```
Please enter the ISBN of book:
ZW100004
find following books
Book name: Gone with the wind
Author name: Margaret Mitchell
ISBN: ZW100004
Subject: Novel
Loan type: normal
Shelf: A1
Inventoryt: 3
Lend number: 0
Note: Modify old book to new book
Please enter any key to continue...
```

Modify successful

Delete Users function:

For testing this function, register a account at the beginning and record the details about this account, then use Delete Users function delete this account, finally, try to log in system use this account confirm it.

```
Username: fordelete
Type: student
No history.
Enter any key to return last menu.
```

Account: fordelete Password: 123

```
Please enter the username which will be deleted:
fordelete
Delete user successfully!
Enter any key to continue...
```

Use Delete User function

```
Please enter your username:
fordelete
This username does not exist! Please enter again.
Please enter your username:
```

Delete successful

```
Please enter the username which will be deleted:
0000
This username does not exist! Please enter again!
Please enter the username which will be deleted:
```

The function will give notice to user when this user account are not exist

View all boos on loan function:

For testing this function, loan a book at the beginning, then use View all books on loan function.

```
There is not any book on loan.
Enter any key to continue...
```

When there are no books on loan, this function will give notice to user.

```
Please enter the book'ISBN which you want to borrow:
ZW100000
find this book
Book name: Jane Eyre
Author name: Charlotte Bronte
ISBN: ZW100000
Inventoryt: 1
Lend number: 0
Note: No

Do you want to borrow this book?
1-Yes and 2-No
1
borrow book successfully!
You are a student and you can keep this book for 30 days. Please return it on ti
me.
Please enter any key to continue...
```

Borrow one book

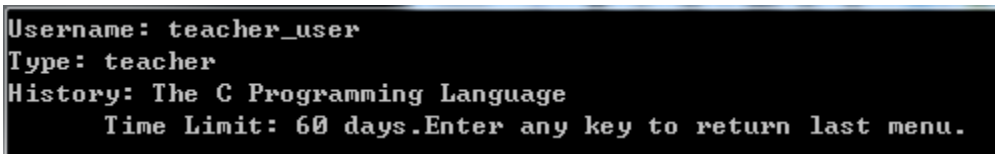
```
find borrowed book
Book name: Jane Eyre
ISBN: ZW100000
Inventoryt: 0
Lend number: 1
Enter any key to continue...
```

Use this function and display the books on loan

Log in service menu:

When log in successful, this service menu will display

My information function



User could check their user name, user type and borrow book history

Search book function:

Search book menu

```
Please enter the name of book:
Jane Eyre
find following books
Book name: Jane Eyre
Author name: Charlotte Bronte
ISBN: ZW100000
Subject: Novel
Loan type: normal
Shelf: A1
Inventoryt: 1
Lend number: 0
Note: No
Please enter any key to continue...
```

Use book name to search

```
Please enter the name of book:
0000
this book is not in the library
Please enter any key to continue...
```

When this book are not in the library

```
Please enter the author of book:
Brian W. Kernighan & Dennis M. Ritchie
find following books
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Subject: EEE101
Loan type: normal
Shelf: A2
Inventoryt: 2
Lend number: 0
Note: No
Please enter any key to continue...
```

Use book author to search

```
Please enter the author of book:
0000
this book is not in the library
Please enter any key to continue...
```

When this book are not in the library

```
Please enter the ISBN of book:
ZW100004
find following books
Book name: Gone with the wind
Author name: Margaret Mitchell
ISBN: ZW100004
Subject: Novel
Loan type: normal
Shelf: A1
Inventoryt: 2
Lend number: 0
Note: Modify old book to new book
Please enter any key to continue...
```

Use book ISBN to search

```
Please enter the ISBN of book:
0000
this book is not in the library
Please enter any key to continue...
```

When this book are not in the library

```
Please enter the subject of book:
EEE101
find following books
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Subject: EEE101
Loan type: normal
Shelf: A2
Inventoryt: 2
Lend number: 0
Note: No
Please enter any key to continue...
```

Use subject to search

```
Please enter the subject of book:
Novel
find following books
Book name: Jane Eyre
Author name: Charlotte Bronte
ISBN: ZW100000
Subject: Novel
Loan type: normal
Shelf: A1
Inventoryt: 1
Lend number: 0
Note: No
find following books
Book name: Gone with the wind
Author name: Margaret Mitchell
ISBN: ZW100004
Subject: Novel
Loan type: normal
Shelf: A1
Inventoryt: 2
Lend number: 0
Note: Modify old book to new book
Please enter any key to continue...
```

Use subject to search

```
Please enter the subject of book:
0000
this book is not in the library
Please enter any key to continue...
```

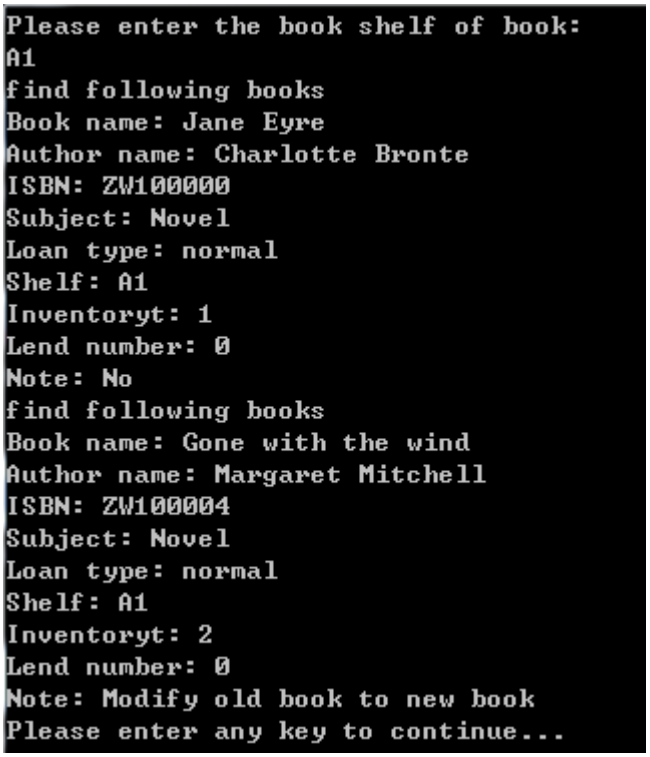
When this subject not in the library

```
Please enter the loan type of book:
normal
find following books
Book name: Jane Eyre
Author name: Charlotte Bronte
ISBN: ZW100000
Subject: Novel
Loan type: normal
Shelf: A1
Inventoryt: 1
Lend number: 0
Note: No
find following books
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Subject: EEE101
Loan type: normal
Shelf: A2
Inventoryt: 2
Lend number: 0
Note: No
find following books
Book name: Gone with the wind
Author name: Margaret Mitchell
ISBN: ZW100004
Subject: Novel
Loan type: normal
Shelf: A1
Inventoryt: 2
Lend number: 0
Note: Modify old book to new book
Please enter any key to continue...
```

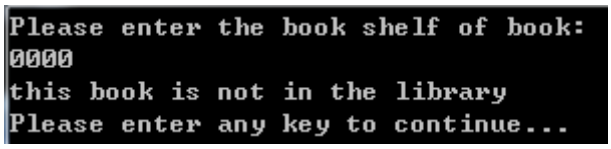
Use loan type to search

```
Please enter the loan type of book:
0000
this book is not in the library
Please enter any key to continue...
```

When this type not in the library



Use book shelf to search



When this library do not have this shelf

Borrow book function:

For testing this function, search a book at the beginning and record the inventory of this book, then borrow this book, finally, search it again to compare the inventory and lend number.

```
Please enter the ISBN of book:
ZW100001
find following books
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Subject: EEE101
Loan type: normal
Shelf: A2
Inventoryt: 2
Lend number: 0
Note: No
Please enter any key to continue...
```

Inventory is 2

```
Please enter the book'ISBN which you want to borrow:
ZW100001
find this book
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Inventoryt: 2
Lend number: 0
Note: No

Do you want to borrow this book?
1-Yes and 2-No
1
borrow book successfully!
You are a student and you can keep this book for 30 days. Please return it on ti
me.
Please enter any key to continue...
```

Borrow successful (student account)

```
Please enter the ISBN of book:
ZW100001
find following books
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Subject: EEE101
Loan type: normal
Shelf: A2
Inventory: 1
Lend number: 1
Note: No
Please enter any key to continue...
```

Inventory is 1 and land number is 1

```
Please enter the book'ISBN which you want to borrow:
ZW100001
find this book
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Inventory: 1
Lend number: 1
Note: No

Do you want to borrow this book?
1-Yes and 2-No
1
borrow book successfully!
You are a teacher and you can keep this book for 60 days. Please return it on ti
me.
Please enter any key to continue...
```

Borrow successful (teacher account)

```
Please enter the ISBN of book:
ZW100001
find following books
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Subject: EEE101
Loan type: normal
Shelf: A2
Inventory: 0
Lend number: 2
Note: No
Please enter any key to continue...
```

Inventory is 0 and land number is 2

```
Please enter the book'ISBN which you want to borrow:
ZW100001
find this book
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Inventory: 0
Lend number: 2
Note: No
Sorry, this book borrowed out
Please enter any key to continue...
```

When inventory equal to 0, user cannot borrow this book

Return book function:

For testing this function, search a book which borrowed by user at the beginning and record the lend number of this book, then return this book, finally, search it again to compare the inventory and lend number.

```
Username: teacher_user
Type: teacher
History: The C Programming Language
      Time Limit: 60 days.Enter any key to return last menu.
```

```
Please enter the book'ISBN which you want to return:
ZW100001
find this book
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Inventory: 0
Lend number: 2
Note: No

Do you want to return this book?
1-Yes and 2-No
1
return book successfully!
Please enter any key to continue...
```

Return successful

```
Please enter the ISBN of book:
ZW100001
find following books
Book name: The C Programming Language
Author name: Brian W. Kernighan & Dennis M. Ritchie
ISBN: ZW100001
Subject: EEE101
Loan type: normal
Shelf: A2
Inventory: 1
Lend number: 1
Note: No
Please enter any key to continue...
```

Inventory is 1 and lend number is 1

View all available book function:

```
find unborrowed book
Book name: Jane Eyre
ISBN: ZW100000
Inventoryt: 1
Lend number: 0

find unborrowed book
Book name: The C Programming Language
ISBN: ZW100001
Inventoryt: 1
Lend number: 1

find unborrowed book
Book name: test
ISBN: ZW100002
Inventoryt: 2
Lend number: 0

find unborrowed book
Book name: Gone with the wind
ISBN: ZW100004
Inventoryt: 2
Lend number: 0
```

This function will display all available books

```
There is not any book left.
```

If there are not available books, this function will give notice to user

ALL