

**19BIT0292**

**Bhaumik Tandan**

**DIGITAL ASSIGNMENT-2**

**DATA STRUCTURES**

**AND**

**ALGORITHMS**

**LABORATORY**

CSE2011

L57+L58

**Q1)** Create a linked list Swap two adjacent elements by

(a) Interchanging the elements itself.

(b) Adjusting only the pointers (and not data)

In both cases, assume that elements are stored in single linked list and also with double linked list.

(a) swap\_item.h

**CODE**

#pragma once

void swap\_item(struct node\* p,struct node\* q)

{

int t=p->d;

p->d=q->d;

q->d=t;

}

(b) swap\_pointer.h

**CODE**

#pragma once

void swap\_next(struct node\* p,struct node\* q)

{

struct node\* t =q->n;

q->n=p->n;

p->n=t;

}

void swap\_pointer(struct node\* p,struct node\* q)

{

struct node\* t=p->n;

p->n=q->n;

q->n=t;

swap\_next(p->n,q->n);

}

void swap\_pointer\_change\_head(struct node\* q)

{

struct node\* t=h;

h=q->n;

q->n=t;

swap\_next(h,q->n);

}

swap\_pointer\_dll.h

#pragma once

void swap\_prev(struct node \*p,struct node \*q)

{

struct node\* t =q->p;

q->p=p->p;

p->p=t;

}

void swap\_pointer\_dll(struct node \*p,struct node \*q)

{

q->p->n=p;

if(p==h)

h=q;

else

p->p->n=q;

if(q->n)

q->n->p=p;

p->n->p=q;

swap\_next(p,q);

swap\_prev(p,q);

}

**single\_ll.c**

**CODE**

#include "ll.h"//[**Click here for the souce code of ll.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/ll.h)

#include "swap\_item.h"//[**Click here for the souce code of swap\_item.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/swap_item.h)

#include "swap\_pointer.h"//[**Click here for the source code of swap\_pointer.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/swap_pointer.h)

#include "get\_node.h"//[**Click here for the souce code of get\_node.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/get_node.h)

#include "swap.h"//[**Click here for the souce code of swap.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/swap.h)

#include "menu.h"//[**Click here for the souce code of menu.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/menu.h)

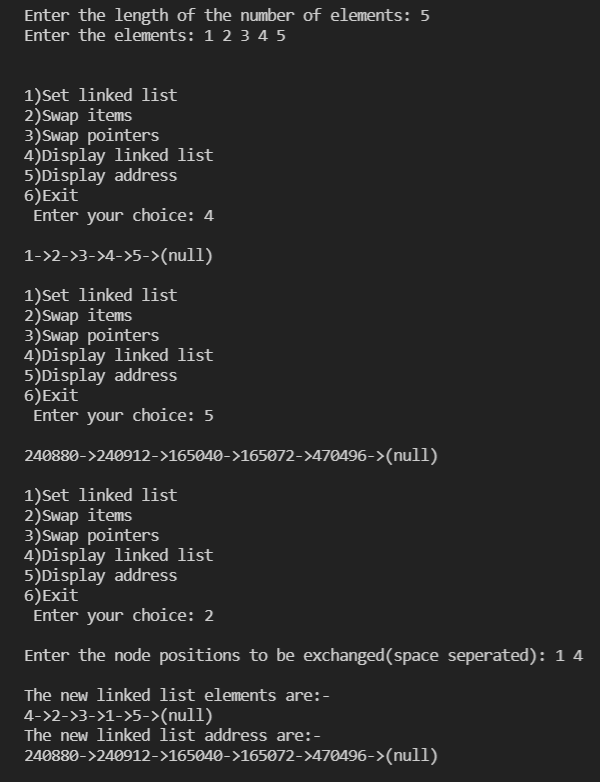
main()

{

takeinput();

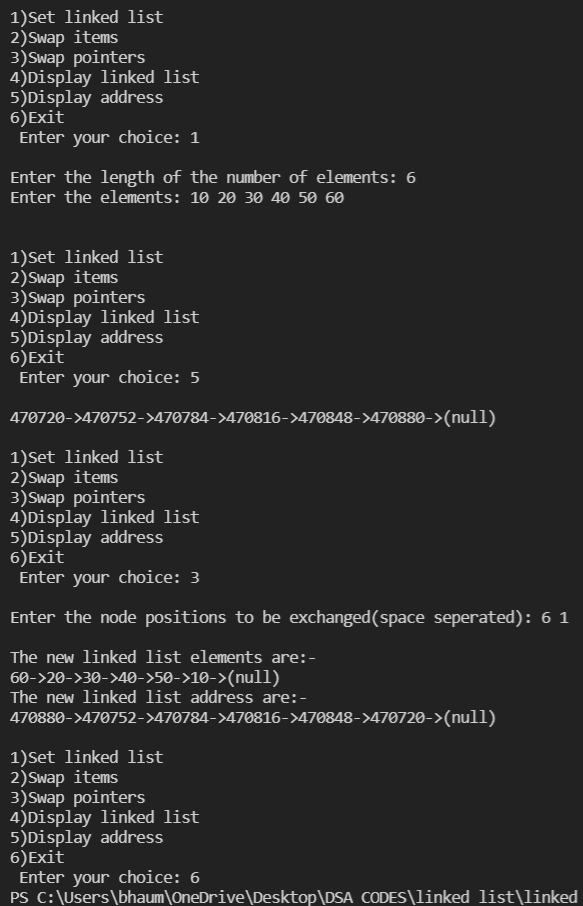
menu();

}

**OUTPUT**

**Swapping items**

**Swapping pointers**

****

**double\_ll.c**

**CODE**

#include "../doubly\_linked\_list.h"//[**Click here for the souce code of doubly\_linked\_list.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/doubly_linked_list.h)

#include "swap\_item.h"//[**Click here for the souce code of swap\_item.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/swap_item.h)

#include "swap\_pointer.h"**//**[**Click here for the source code of swap\_pointer.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/swap_pointer.h)

#include "get\_node.h"**//**[**Click here for the souce code of get\_node.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/get_node.h)

#include "swap\_pointer\_dll.h"//[**Click here for the souce code of swap\_pointer\_dll.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/swap_pointer_dll.h)

#include "swap.h"//[**Click here for the souce code of swap.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/swap.h)

#include "menu.h"//[**Click here for the souce code of menu.h**](https://github.com/Bhaumik-Tandan/DSA_CODES/blob/master/linked list/linked_list_swap/menu.h)

main()

{

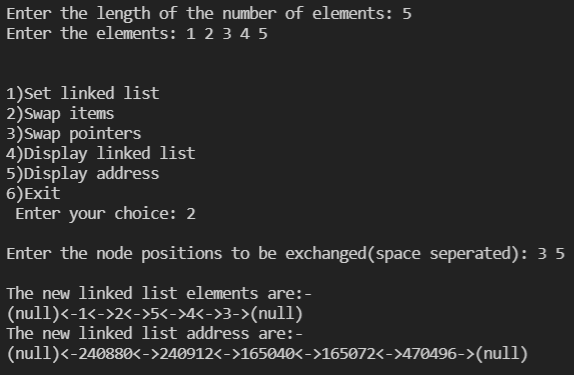
takeinput();

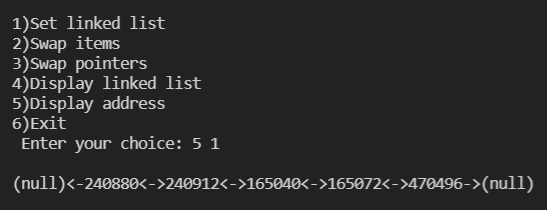
menu();

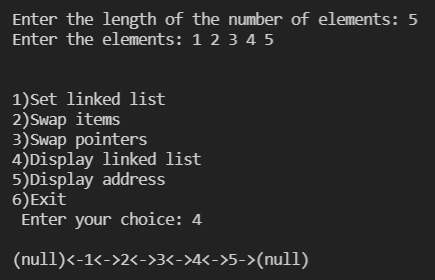
}

**OUTPUT**

**Swapping items**

**Swapping pointers**

**Display linked list**

****

# [**CLICK HERE FOR GITHUB LINK OF WHOLE SOURCE CODE**](https://github.com/Bhaumik-Tandan/DSA_CODES/tree/master/linked list/linked_list_swap)

**Q2)**Given single circular linked list containing a set of data. Obtain the following from this data structure:

(a) Reverse the direction of links

(b) Forgiven two elements in the list, find the distance (number of nodes) between them.

**cll.h**

**CODE**

#include<stdio.h>

#include<stdlib.h>

#pragma once

struct node

{

int d;

struct node \*n;

} \* l;

typedef struct node node;

void ins(int n)

{

node \*t = (node \*)malloc(sizeof(node));

t->d = n;

if (!l)

{

l = t;

t->n = l;

return;

}

t->n=l->n;

l->n=t;

l=t;

}

void disp()

{

printf("\n-");

node \*t = l;

int i=0;

do

{

t = t->n;

printf("->%d",t->d);

i++;

}

while(t!=l);

printf("--\n|");

for(int j=0;j<i\*3+2;j++)

printf("\_");

printf("|");

}

void takeinput()

{

int n;

printf("\nEnter the number of elements: ");

scanf("%d",&n);

printf("Enter the elements: ");

for (int i = 0; i < n; i++)

{

int t;

scanf("%d",&t);

ins(t);

}

}

(a) reverse.h

**CODE**

#include "cll.h"

void reverse()

{

if(!l)

{

printf("\nList empty");

return;

}

node \*prev=l,\*c=l->n;

l=c;

do

{

node\* t=c->n;

c->n=prev;

prev=c;

c=t;

}

while(c!=l);

}

main()

{

takeinput();

disp();

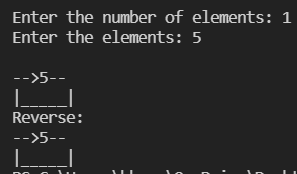
reverse();

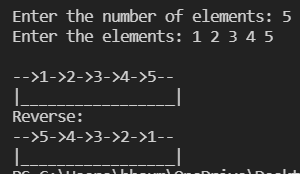
printf("\nReverse: ");

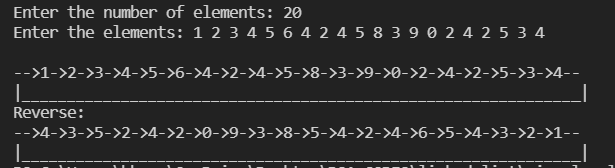
disp();

}

**OUTPUT**







(b) distance.h

**CODE**

#include "cll.h"

void distance()

{

if(!l)

{

printf("\nList empty");

return;

}

printf("\nEnter the first element: ");

int f,s;

scanf("%d",&f);

printf("Enter the second element: ");

scanf("%d",&s);

node\* p=l;

while(p->d!=f)

p=p->n;

f=0;

while(p->d!=s && ++f)//used to increment as well && does not goes to 2nd if first found false

p=p->n;

printf("Distance=%d",f);

}

main()

{

takeinput();

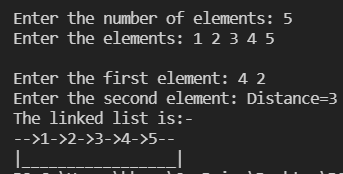
distance();

printf("\nThe linked list is:-");

disp();

}

**OUTPUT**



# [**CLICK HERE FOR GITHUB LINK OF WHOLE SOURCE CODE**](https://github.com/Bhaumik-Tandan/DSA_CODES/tree/master/linked list/circular_ll)

**Q3)**It is required to maintain a library database using a number of lists as mentioned: The list BOOKS contains the information like title, accession number, and tag field (to indicate whether a book is issued or not) for all the books in a library. Note that each book can be there in multiple copies, but there accession numbers are different. Another list SUBSCRIBERS will contain the name, borrower number and the list of books (with date of issues) he has issued. Assume that a subscriber can issue up to five books at the most and no two copies of the same book.

(a) Design a suitable data structure using single linked list.

**header\_function.h**

**CODE**

#pragma once

#include<stdbool.h>

#include<string.h>

#include <termios.h>

#include <unistd.h>

struct book\_node

{

char title[30];

int accession\_number;

bool tag;

struct book\_node \*n;

}\*head\_book;

typedef struct book\_node book\_node;

struct subscriber\_node

{

char name[30];

int borrower\_number;

char password[30];

book\_node \*book[5];

char date[5][20];

int number;

struct subscriber\_node \*n;

}\*head\_subscriber;

typedef struct subscriber\_node subscriber\_node;

//global functions

void admin();

void borrower();

void print\_book\_direct(subscriber\_node\* p);

void print\_book(int borrower\_number);

void disp\_avail\_book();

void print\_subscriber(char book[30]);

void issue\_directly(subscriber\_node\* p,char book[30],char date[30]);

void issue\_book(char book[30],int borrower\_number,char date[30],char pas[30]);

void return\_directly(subscriber\_node\* p,char book[30]);

void return\_book(char book[30],int borrower\_number,char pas[30]);

void add\_book(char title[30],int accession\_number);

book\_node\* set\_tag(char book[30],bool tag);

void home();

subscriber\_node\* subscriber\_exists(int borrower\_number,char password[30]);

void add\_subscriber(char name[30],int borrower\_number,char password[30]);

#include "book.h"

#include "suscriber.h"

#include "./operation/return.h"

#include "./operation/issue.h"

#include "./operation/display/print\_books.h"

#include "./operation/display/subscriber\_of\_book.h"

void \_\_attribute\_\_((constructor)) setup();

#include "./default\_opearations/setup.h"//contains default operation

**book.h**

**CODE**

#pragma once

void add\_book(char title[30],int accession\_number)

{

book\_node \*t=(book\_node\*)malloc(sizeof(book\_node));

strcpy(t->title,title);

t->accession\_number=accession\_number;

t->tag=0;

t->n=0;

if(!head\_book)

{

head\_book=t;

printf("\n Book \"%s\" added with accession number %d",title,accession\_number);

return;

}

book\_node \*p=head\_book;

while(p->n)

if(p->accession\_number==accession\_number)//check if the accession number is unique or not

{

printf("\n Accession number %d already assigned with book \"%s\"",accession\_number,p->title);

return;

}

else

p=p->n;

if(p->accession\_number==accession\_number)

{

printf("\n Accession number %d already assigned with book \"%s\"",accession\_number,p->title);

return;

}

printf("\n Book \"%s\" added with accession number %d",title,accession\_number);

p->n=t;

}

book\_node\* set\_tag(char book[30],bool tag)//return 1 on successful change

{

book\_node \*p=head\_book;

while(p && !(strcmp(p->title,book)==0 && p->tag!=tag))//find the book wit book number and not isssued

p=p->n;

if(!p)

return 0;

p->tag=tag;

return p;

}

**subscriber.h**

**CODE**

#pragma once

subscriber\_node\* subscriber\_exists(int borrower\_number,char password[30])

{

subscriber\_node \*p=head\_subscriber;

while(p && !(p->borrower\_number==borrower\_number && strcmp(p->password,password)==0))

p=p->n;

return p;

}

void add\_subscriber(char name[30],int borrower\_number,char password[30])

{

subscriber\_node \*t=(subscriber\_node\*)malloc(sizeof(subscriber\_node));

strcpy(t->name,name);

t->borrower\_number=borrower\_number;

strcpy(t->password,password);

t->number=0;

t->n=0;

if(!head\_subscriber)

{

head\_subscriber=t;

printf("\n Borrower %s with borrower number %d added",name,borrower\_number);

return;

}

subscriber\_node \*p=head\_subscriber;

while(p->n)

if(p->borrower\_number==borrower\_number)//check if the borrower number is unique or not

{

printf("\n Borrower number %d already exists with %s",borrower\_number,p->name);

return;

}

else

p=p->n;

if(p->borrower\_number==borrower\_number)

{

printf("\n Borrower number %d already exists with %s",borrower\_number,p->name);

return;

}

printf("\n Borrower %s with borrower number %d added",name,borrower\_number);

p->n=t;

}

(b) Write a menu driven program using C for the following:

(i) To issue a book

**/operation/issue.h**

**CODE**

#pragma once

void issue\_directly(subscriber\_node\* p,char book[30],char date[20])

{

if(p->number==5)

{

printf("\n %s has exceeded the limit of issuing books",p->name);//indicate limit exceeds

return;

}

for(int i=0;i<p->number;i++)

if(strcmp(p->book[i]->title,book)==0)// indicate book already issued

{

printf("\n %s has already issued %s with accession number %d",p->name,book,p->book[i]->accession\_number);

return;//indicate book already issued

}

book\_node \*issue=set\_tag(book,1);

if(!issue)

{

printf("\n %s is not available in the library",book);

return;

}

int n=p->number;

p->number=n+1;

p->book[n]=issue;

strcpy(p->date[n],date);

printf("\n %s with accession number %d has been issued to %s with borrower number %d on %s",book,p->book[n]->accession\_number,p->name,p->borrower\_number,date);

}

void issue\_book(char book[30],int borrower\_number,char date[10],char pas[30])

{

subscriber\_node \*p=subscriber\_exists(borrower\_number,pas);

if(!p)

printf("\nWrong credentials");

else

issue\_directly(p,book,date);

}

(ii) Return a book

**/operation/return.h**

**CODE**

#pragma once

void return\_directly(subscriber\_node\* p,char book[30])

{

if(p->number==0)

{

printf("\n %d) %s has no pending returns",p->borrower\_number,p->name);//indicate no book

return;

}

int f=-1;

for(int i=0;i<p->number;i++)

if(strcmp(p->book[i]->title,book)==0)// indicate book issued

{

f=i;

break;

}

if(f==-1)

{

printf("\n %d) %s has not issued book %s",p->borrower\_number,p->name,book);//indicate no book

return;

}

p->book[f]->tag=0;//set to not issued

printf("\n %s with accession number %d has been returned by %s with borrower number %d which was issued on %s",

book,p->book[f]->accession\_number,p->name,p->borrower\_number,p->date[f]);

p->book[f]=0;

for(int i=f;i<p->number-1;i++)

{

p->book[i]=p->book[i+1];

strcpy(p->date[i],p->date[i+1]);

}

p->number-=1;

}

void return\_book(char book[30],int borrower\_number,char pas[30])

{

subscriber\_node \*p=subscriber\_exists(borrower\_number,pas);//imported in login

if(!p)

printf("\nWrong credentials");

else

return\_directly(p,book);

}

(iii) Show the list of books issued by a subscriber

**/operation/display/print\_book.h**

**CODE**

#pragma once

void print\_book\_direct(subscriber\_node\* p)

{

if(p->number==0)

{

printf("\n %d) %s has not issued any book",p->borrower\_number,p->name);

return;//indicate book not found

}

printf("\n Books issued by %s are: ",p->name);

for(int i=0;i<p->number;i++)

printf("\n %d) %s on %s",p->book[i]->accession\_number,p->book[i]->title,p->date[i]);

}

void print\_book(int borrower\_number)

{

subscriber\_node \*p=head\_subscriber;

printf("\n");

while(p && p->borrower\_number!=borrower\_number)

p=p->n;

if(!p)

printf("\n %d does not exists in the record",borrower\_number);

else

print\_book\_direct(p);

}

void disp\_avail\_book()

{

bool f=0;

printf("\n\n Available books in the library are:-");

book\_node \*p=head\_book;

while(p)

{

if(p->tag==0)

{

printf("\n %d)%s",p->accession\_number,p->title);

f=1;

}

p=p->n;

}

if(!f)

printf("\n No books available");

}

(iv) Given a title, find out to whom it has issued.

**/operation/display/subscriber\_of\_book.h**

**CODE**

#pragma once

void print\_subscriber(char book[30])

{

subscriber\_node \*p=head\_subscriber;

printf("\n\n %s has been issued by: ",book);

int f=-1;

while(p)

{

for(int i=0;i<p->number;i++)

if(!strcmp(p->book[i]->title,book))

{

printf("\n %d) %s on %s: %d",p->borrower\_number,p->name,p->date[i],p->book[i]->accession\_number);

f=1;

break;

}

p=p->n;

}

if(f==-1)

printf("\n No borrowers found");

}

**For the purpose of test the bellow code has been added**

**default\_operations/setup.h**

#pragma once

void setup() {//it runs automatically

#include "header.h"

add\_default\_subscribers();

add\_default\_books();

add\_default\_issue();

// bellow all are just for testing

test\_return\_operations();

test\_display\_book\_operation();

test\_subscriber\_disp\_operation();

printf("\n---------------------Default operations ended------------------------");

}

**default\_operations/subscriber.h**

#pragma once

void add\_default\_subscribers()

{

printf("\n");

add\_subscriber("Rohan Gupta",221,"1234");

add\_subscriber("Bhaumik Tandan",292,"1234");

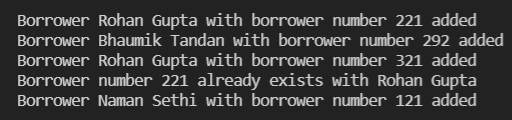
add\_subscriber("Rohan Gupta",321,"1234");

add\_subscriber("Rahul Misra",221,"1234");//will not get added

add\_subscriber("Naman Sethi",121,"1234");

}

Duplicate borrower number will not get added



**default\_operations/books.h**

#pragma once

void add\_default\_books()

{

printf("\n");

add\_book("Three men in a boat",1564);

add\_book("7 ages",9564);

add\_book("Macbeth",3253);

add\_book("End game",1014);

add\_book("Dance of dragons",1024);

add\_book("Invisible Man",1114);

add\_book("Dead end",1814);

add\_book("Clash of Kings",7814);

add\_book("Three men in a boat",1564);//will not get added

add\_book("7 ages",1264);//will get added as accession number changed

add\_book("Three men in a boat",1253);

add\_book("7 ages",1164);

add\_book("End game",1019);

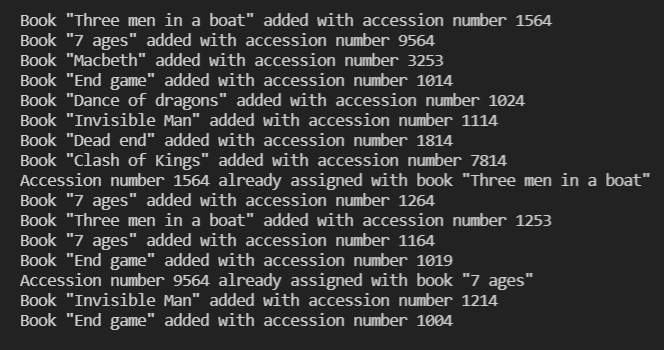
add\_book("Macbeth",9564);//will not get added

add\_book("Invisible Man",1214);

add\_book("End game",1004);

}

Duplicate accession number will not get added



**default\_operations/issue.h**

#pragma once

void add\_default\_issue()

{

printf("\n");

issue\_book("Macbeth",221,"12/2/2013","1234");

issue\_book("Three men in a boat",221,"13/3/2013","1234");

issue\_book("7 ages",221,"13/3/2013","1234");

issue\_book("Invisible Man",221,"13/9/2013","1234");

issue\_book("Three men in a boat",221,"13/3/2013","1234");//will not get issued as already issued

issue\_book("Three men in a boat",292,"13/3/2013","1234");//will get issued by different person

issue\_book("Dead end",292,"23/3/2013","1234");

issue\_book("End game",221,"13/3/2014","1234");

issue\_book("Clash of Kings",221,"23/3/2013","1234");//will not get issued as limit exeded

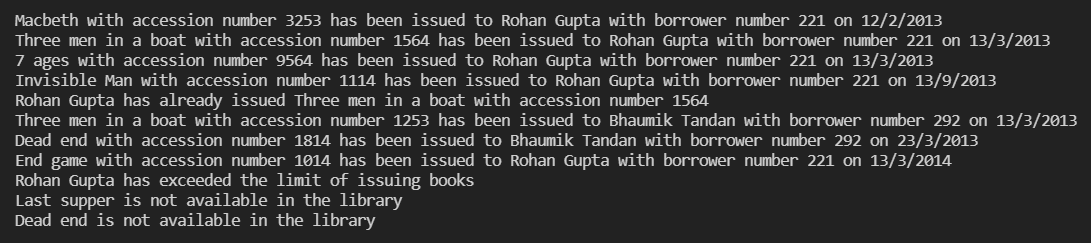
issue\_book("Last supper",292,"23/3/2013","1234");//will not get issued as book not available

issue\_book("Dead end",321,"23/3/2013","1234");//will not get issued as book not available as we only had one copy already issued by 292

}

Out of stock book will not get issued, than 5 books will not get issued and

the book which is not in library record will show not available



**default\_operations/return.h**

#pragma once

void test\_return\_operations()

{

printf("\n");

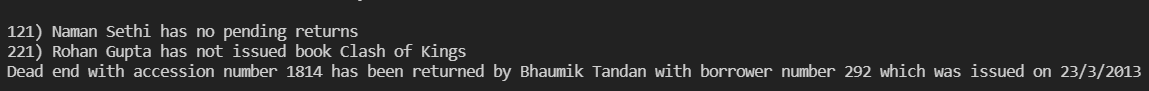
return\_book("Three men in a boat",121,"1234");//will not return as he has not issued any book

return\_book("Clash of Kings",221,"1234");//will not return as he has not issued the given book

return\_book("Dead end",292,"1234");

}

Borrower which does not have pending return and borrowers which do not issue that particular book will display message



**default\_operations/disp\_book.h**

#pragma once

void test\_display\_book\_operation()

{

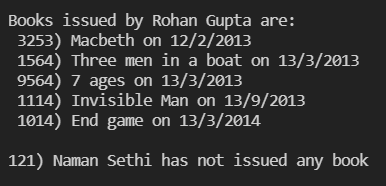
printf("\n");

print\_book(221);

print\_book(121);//will print not issued message

}

User who has not issued any book message will be displayed



**default\_operations/disp\_subscriber.h**

#pragma once

void test\_subscriber\_disp\_operation()

{

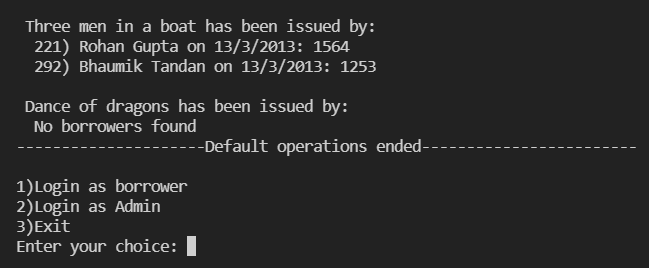
printf("\n");

print\_subscriber("Three men in a boat");

print\_subscriber("Dance of dragons");

}

Books that does not have subscriber then menu gets displayed



**Menu demo**

# 

# 

# 

# 

# 

# 

# 

# [**CLICK HERE FOR GITHUB LINK OF WHOLE SOURCE CODE**](https://github.com/Bhaumik-Tandan/Library_Management_System)