COMPUTER SCIENCE PROJECT FILE

LIBRARY MANAGEMENT SYSTEM

PROJECT BY –

SHRUTI PRABHU

XII B

DELHI PUBLIC SCHOOL

DELHI PUBLIC SCHOOL
NERUL , NAVI MUMBAI

Certificate
Session 2020-2021

This is to certify that SHRUTI PRABHU of Class XII ,

Board Roll Number____15608142______

has satisfactorily completed the Computer Science Project (Subject code – 083) on

Topic: Library Management System

in partial fulfillment of the class 12th curriculum of AISSCE-2021 of the CBSE.

____________	____________	___________	___________
Internal Examiner	**External Examiner**	**Principal sign**	**School Seal**

ACKNOWLEDGEMENT

I hereby thank the school and the staff of the computer department for all the guidance and support extended to me during the course of this project.

My special thanks are due to the Principal for providing us with a right platform and environment to complete the project.

My sincere thanks to Ms. Madhurima Kashyap our Informatics Practices teacher for her able guidance , unflinching support and the affectionate touch without which this project would not have been a success.

Finally I thank all those who have directly or indirectly contributed to the success of this project.

Name : RUJUTA PRAJAKT KULKARNI

Roll No :

INDEX

HARDWARE REQUIREMENT:

i. Printer, to print the required documents of the project.

ii. Ram: 2 GB

iii. Processor: Intel Core i5

SOFTWARE REQUIREMENT:

i. Operating System: Windows 10

ii. Python and MySQL, for execution of the program.

iii. MS Word for presentation of output

ABOUT PHYTHON

Python is a popular general-purpose programming language that can be used for a wide variety of applications. It includes high-level data structures, dynamic typing, dynamic binding, and many more features that make it as useful for complex application development as it is for scripting that connects components together. It can also be extended to make system calls to almost all operating systems and to run code written in C or C++. Due to its ubiquity and ability to run on nearly every system architecture, Python is a universal language found in a variety of different applications.

ABOUT SQL

SQL stands for **Structured Query Language**. SQL lets you access and manipulate databases. SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

The RDBMS that we use today rely on SQL as the engine that allows us to perform all the operations required to create, retrieve, update, and delete data as needed. From an open source perspective these RDBMSs include MySQL, MariaDB, and PostgreSQL as the most commonly used open source RDBMS in production today.

MySQL provides functions to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc.

SYNOPSIS

Library is not just a storage for books, but it’s a place where minds and personalities are shaped. It is a never-ending treasure of knowledge. However, the enormity of a library hinders the error-free operation due to its manual tracking system. Thus creation of an automated program in order to replace physical processes and reduce the chances of human error becomes a viable option.

Library management system aims at developing a tool for smooth functioning of a library by tracking the required data and issuing and returning books which otherwise is done manually, thus saving time and labour. . This project not only provides features to access the titles available in the library but also facilitates the easy user-admin interaction.

AIM OF THE PROJECT

- The Library Management System focuses on effective management of records of the members, books inventory and tracking of the issued and returned books using a user-login feature.

- The program allows users to access the books available in the library based on author’s name, name of the book and genre.

- The user is also provided an option to donate books using the program or remove the record of certain books if required

TABLE STRUCTURES

#1.creating book table

```
'''create table if not exists book(  book_id  int(10) primary key,
book_name varchar(100) unique not null,
author varchar(50) not null,
genre varchar(30) not null,
synopsis varchar(8000),
copies int(5),
copies_available int(5) not null);'''
```

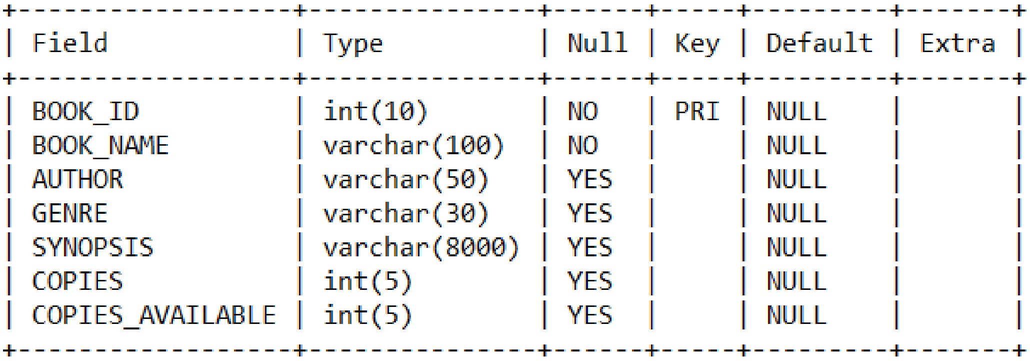
#2.creating table to store member details

```
'''create table if not exists id_password(id int(4) primary key,
name varchar(20) unique not null,
password varchar(20) not null);'''
```

#3.creating table to store details about issued books

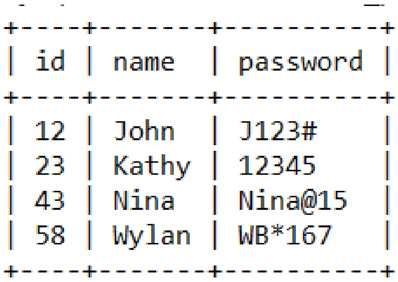
```
'''create table if not exists detail(member varchar(20) not null,
book varchar(100) not null,
status varchar(30) not null);'''
```

Table ‘book’



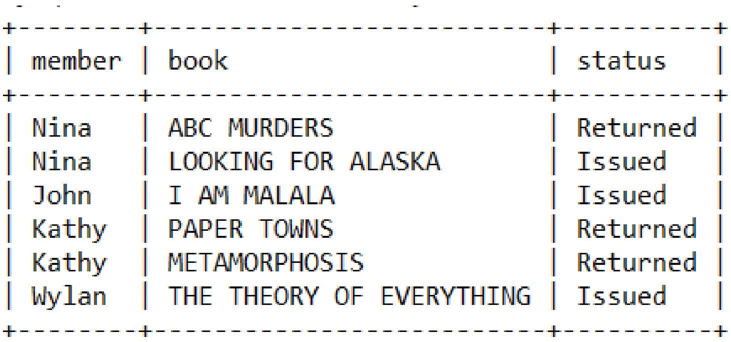
```
+-------------------+----------------+------+-----+---------+-------+
| Field             | Type           | Null | Key | Default | Extra |
+-------------------+----------------+------+-----+---------+-------+
| BOOK_ID           | int(10)        | NO   | PRI | NULL    |       |
| BOOK_NAME         | varchar(100)   | NO   |     | NULL    |       |
| AUTHOR            | varchar(50)    | YES  |     | NULL    |       |
| GENRE             | varchar(30)    | YES  |     | NULL    |       |
| SYNOPSIS          | varchar(8000)  | YES  |     | NULL    |       |
| COPIES            | int(5)         | YES  |     | NULL    |       |
| COPIES_AVAILABLE  | int(5)         | YES  |     | NULL    |       |
+-------------------+----------------+------+-----+---------+-------+
```

Table ‘id_password’



```
+----+-------+----------+
| id | name  | password |
+----+-------+----------+
| 12 | John  | J123#    |
| 23 | Kathy | 12345    |
| 43 | Nina  | Nina@15  |
| 58 | Wylan | WB*167   |
+----+-------+----------+
```

Table ‘detail’



```
+--------+--------------------------+----------+
| member | book                     | status   |
+--------+--------------------------+----------+
| Nina   | ABC MURDERS              | Returned |
| Nina   | LOOKING FOR ALASKA       | Issued   |
| John   | I AM MALALA              | Issued   |
| Kathy  | PAPER TOWNS              | Returned |
| Kathy  | METAMORPHOSIS            | Returned |
| Wylan  | THE THEORY OF EVERYTHING | Issued   |
+--------+--------------------------+----------+
```

CODING

choice()

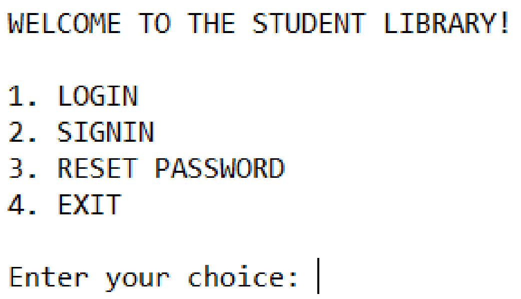
```
def choice():
    print()
    print('WELCOME TO THE STUDENT LIBRARY!')
    a=0
    while a!=4:
        start()
        a=input('Enter your choice: ')
        if a=='1':
            return login()
        elif a=='2':
            signup()
        elif a=='3':
            return password_reset()
        else:
            print('Invalid input.')
            return choice()
```

This function uses the user’s choice to login / sign in / reset password and directs it the respective functions.

start()

```
def start():
    print()
    print('1. LOGIN')
    print('2. SIGNIN')
    print('3. RESET PASSWORD')
    print('4. EXIT')
```

This function displays all the login options available and exit.

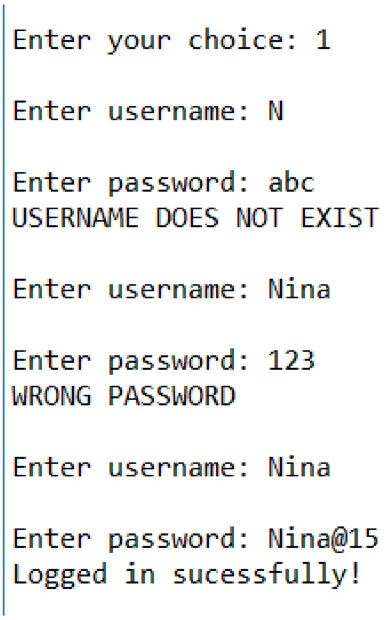


```
WELCOME TO THE STUDENT LIBRARY!

1. LOGIN
2. SIGNIN
3. RESET PASSWORD
4. EXIT

Enter your choice: 
```

login()

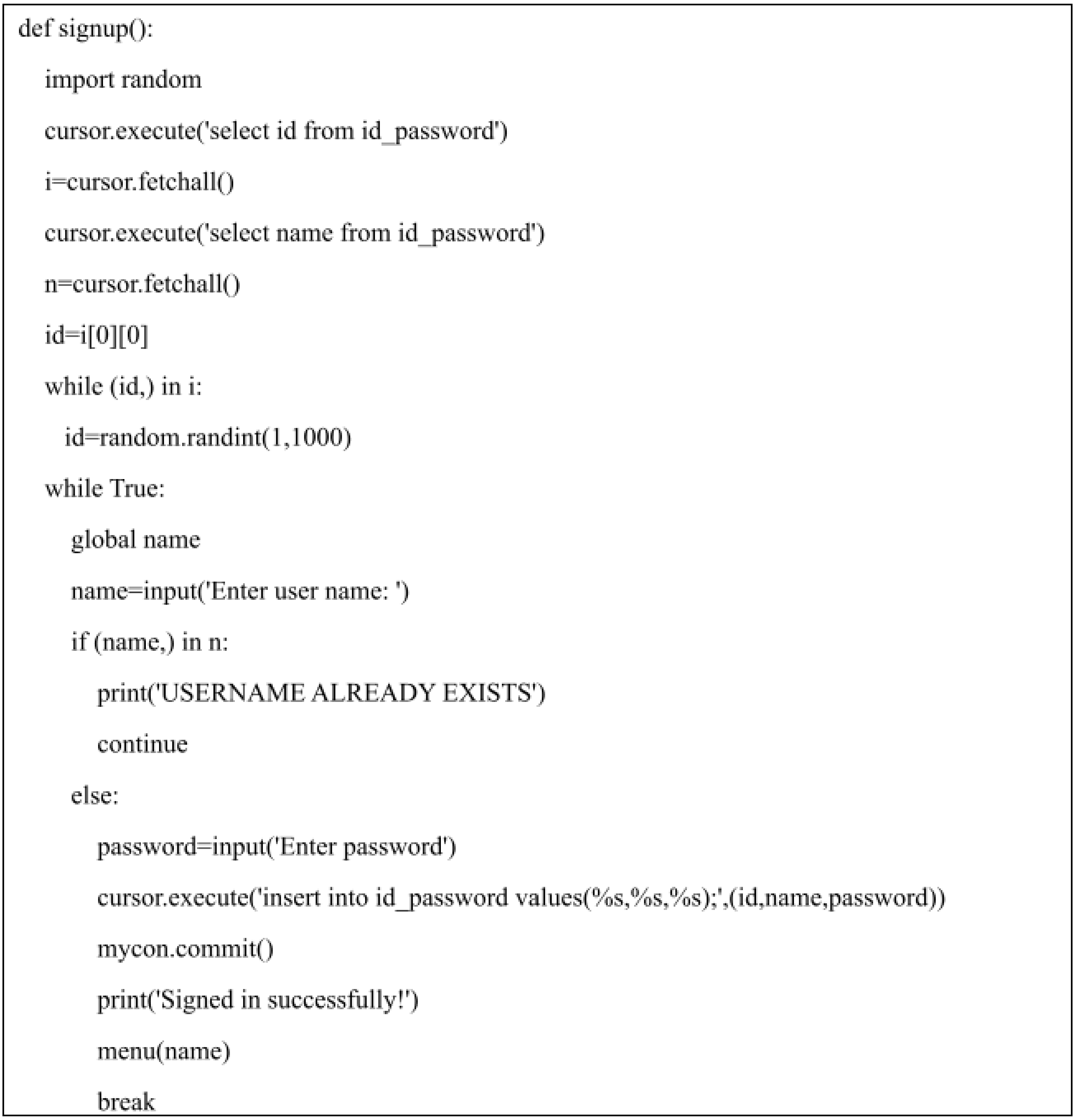
```
def login():
    while True:
        global name
        name=input('Enter username: ')
        password=input('Enter password: ')
        cursor.execute('select * from id_password where name=%s;',(name,))
        l=cursor.fetchall()
        if l!=[]:
            if l[0][2]==password:
                print('Logged in sucessfully!')
                menu(name)
                break
            else:
                print('WRONG PASSWORD')
                continue
        else:
            print('USERNAME DOES NOT EXIST')
            continue
```

This function is used to login into the account after asking for the password. At first username is asked. If the username is present in id_password table in sql it goes on to ask the password, if it is correct, access is granted. Thus, in this way only the admin is guaranteed access.



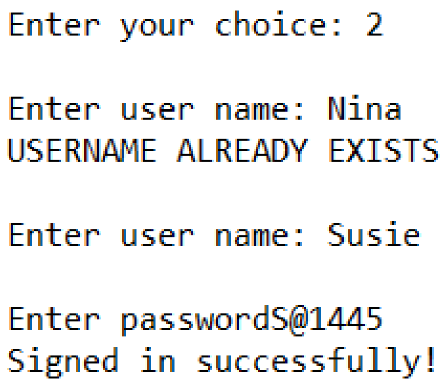
```
Enter your choice: 1

Enter username: N

Enter password: abc
USERNAME DOES NOT EXIST

Enter username: Nina

Enter password: 123
WRONG PASSWORD

Enter username: Nina

Enter password: Nina@15
Logged in sucessfully!
```

signup()



```
def signup():
    import random
    cursor.execute('select id from id_password')
    i=cursor.fetchall()
    cursor.execute('select name from id_password')
    n=cursor.fetchall()
    id=i[0][0]
    while (id,) in i:
        id=random.randint(1,1000)
    while True:
        global name
        name=input('Enter user name: ')
        if (name,) in n:
            print('USERNAME ALREADY EXISTS')
            continue
        else:
            password=input('Enter password')
            cursor.execute('insert into id_password values(%s,%s,%s);',(id,name,password))
            mycon.commit()
            print('Signed in successfully!')
            menu(name)
            break
```

This function allows a new user to create his/her account and hence get a membership in the library. If the user picks a username that already exists then an appropriate message is displayed asking the user to select a different username.

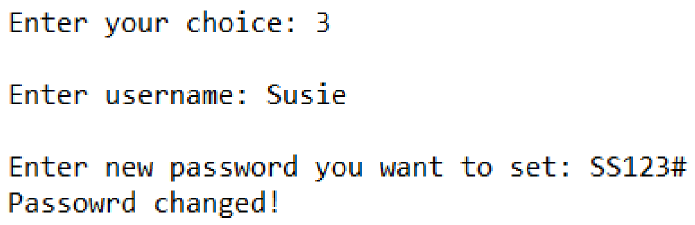


```
Enter your choice: 2

Enter user name: Nina
USERNAME ALREADY EXISTS

Enter user name: Susie

Enter passwordS@1445
Signed in successfully!
```

password_reset()

```
def password_reset():

    while True:

        name=input('Enter username: ')
```

This function allows a user to change his/her password if incase the password has been forgotten. The new password in saved in the id_password table by using the update command.



```
Enter your choice: 3

Enter username: Susie

Enter new password you want to set: SS123#
Passowrd changed!
```

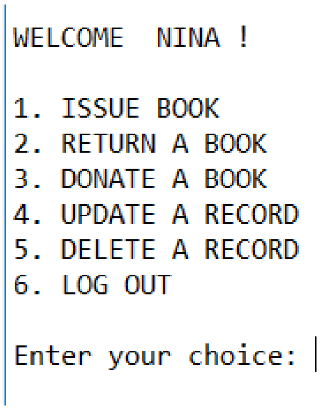
menu(name)

```
def menu(name):
    print()
    c=0
    print("WELCOME ",name.upper(),"!")
    while c!=6:
        print()
        print("1. ISSUE BOOK")
        print("2. RETURN A BOOK")
        print("3. DONATE A BOOK")
        print("4. UPDATE A RECORD")
        print("5. DELETE A RECORD")
        print("6. LOG OUT")
        c=int(input("Enter your choice: "))
        if c>6:
            print("Please enter a valid choice")
        elif c==1:
            print()
```

This function, as the name suggests, displays the main menu of the program. The username is sent as an actual parameter which is used to greet the user in the beginning of the function.

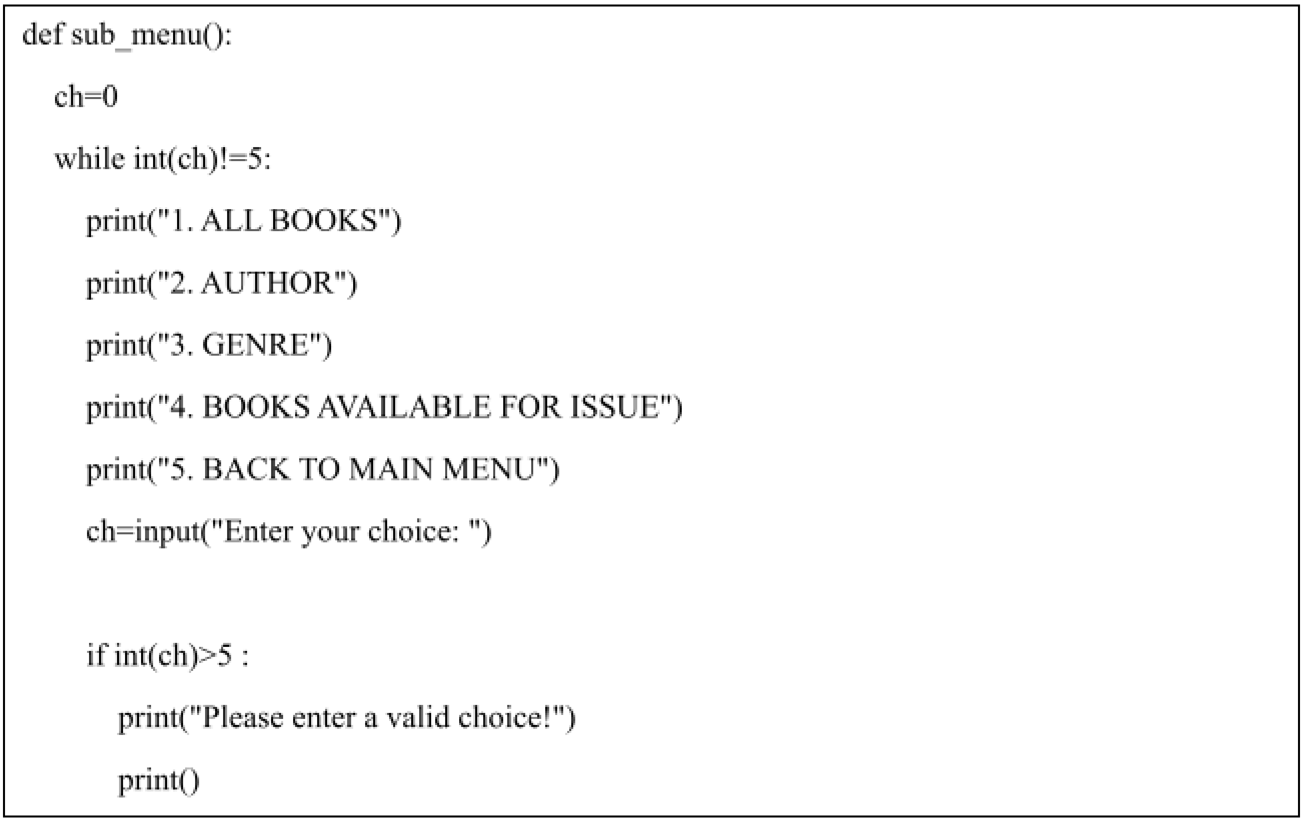
The user is asked to enter a valid input according to whatever he/she wishes to do. If an invalid entry is made then an appropriate message is displayed and the user is asked to enter again.

If a valid entry is made then the user is directed further as the respective functions are called.



```
WELCOME  NINA !

1. ISSUE BOOK
2. RETURN A BOOK
3. DONATE A BOOK
4. UPDATE A RECORD
5. DELETE A RECORD
6. LOG OUT

Enter your choice: 
```

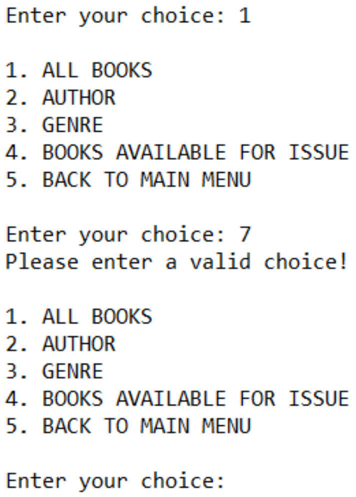
sub_menu()



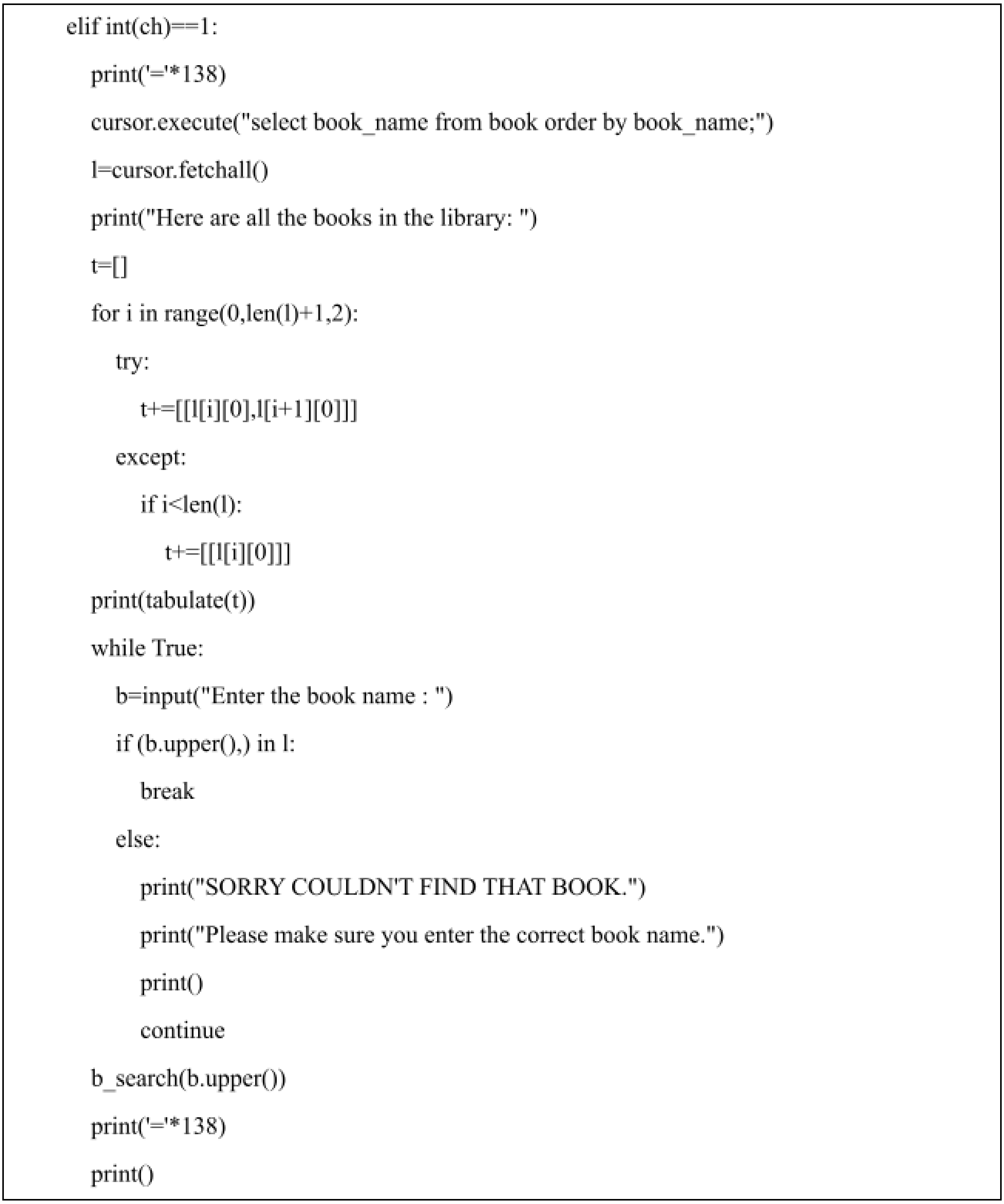
```
def sub_menu():
    ch=0
    while int(ch)!=5:
        print("1. ALL BOOKS")
        print("2. AUTHOR")
        print("3. GENRE")
        print("4. BOOKS AVAILABLE FOR ISSUE")
        print("5. BACK TO MAIN MENU")
        ch=input("Enter your choice: ")

        if int(ch)>5 :
            print("Please enter a valid choice!")
            print()
```

The function sub_menu() offers a menu as shown above on selecting the ‘ISSUE BOOKS’ option in the main menu.

Firstly it checks whether the entered digit is valid to the menu offered or not and displays appropriate message.

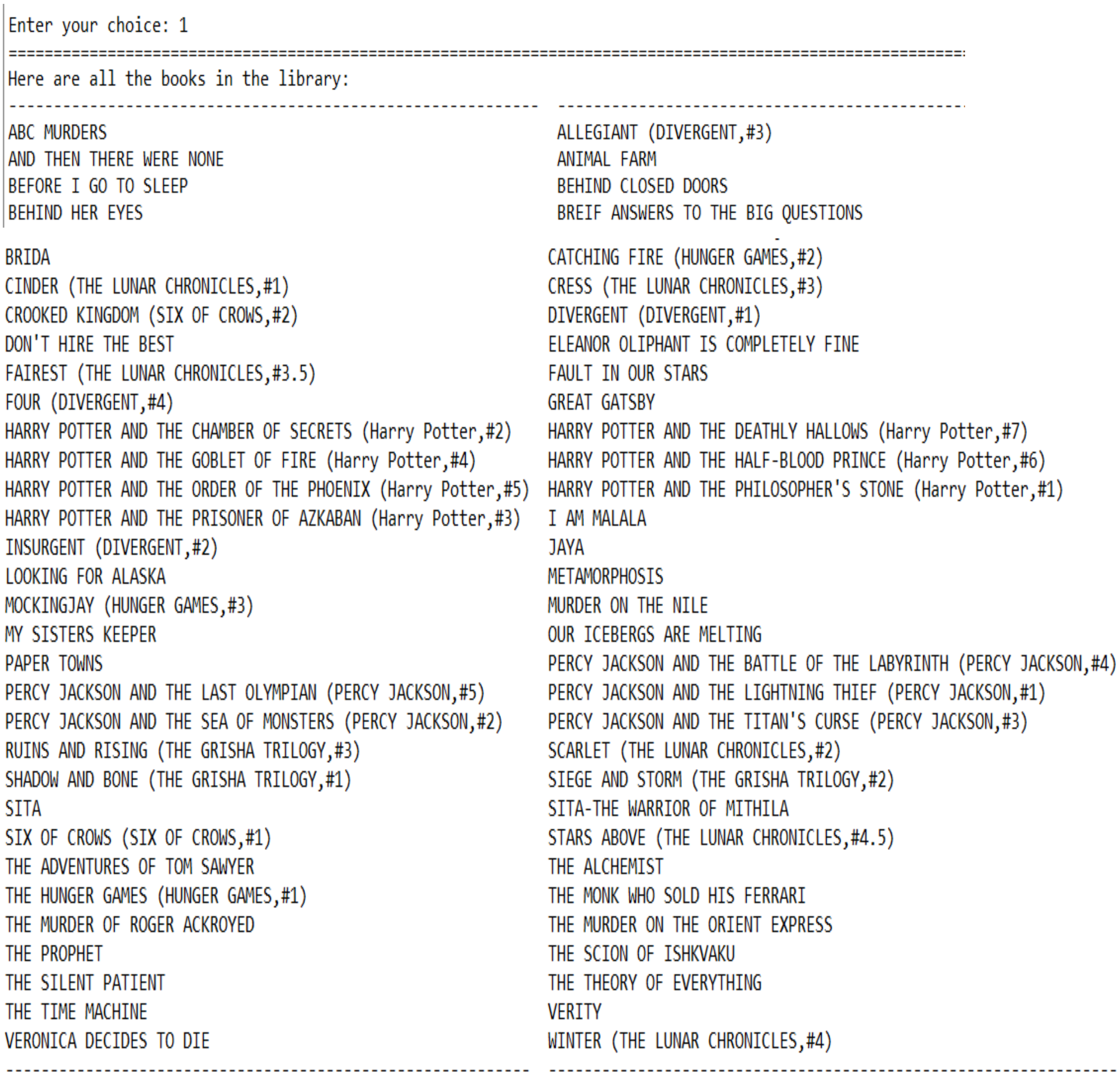


```
Enter your choice: 1

1. ALL BOOKS
2. AUTHOR
3. GENRE
4. BOOKS AVAILABLE FOR ISSUE
5. BACK TO MAIN MENU

Enter your choice: 7
Please enter a valid choice!

1. ALL BOOKS
2. AUTHOR
3. GENRE
4. BOOKS AVAILABLE FOR ISSUE
5. BACK TO MAIN MENU

Enter your choice:
```

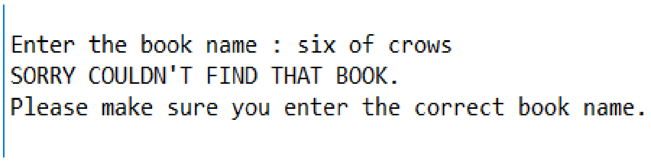


```
elif int(ch)==1:
    print('='*138)
    cursor.execute("select book_name from book order by book_name;")
    l=cursor.fetchall()
    print("Here are all the books in the library: ")
    t=[]
    for i in range(0,len(l)+1,2):
        try:
            t+=[[l[i][0],l[i+1][0]]]
        except:
            if i<len(l):
                t+=[[l[i][0]]]
    print(tabulate(t))
    while True:
        b=input("Enter the book name : ")
        if (b.upper(),) in l:
            break
        else:
            print("SORRY COULDN'T FIND THAT BOOK.")
            print("Please make sure you enter the correct book name.")
            print()
            continue
    b_search(b.upper())
    print('='*138)
    print()
```

When the first option is selected it shows all the books that are in the library in ascending order. The tabulate function allows us to display the data in an organised way. When the name of the book is entered, it checks whether name is a valid entry or not; if it is not then an appropriate message is displayed and the book name is asked to enter again. If the book name entered is correct then the loop breaks and the book name is sent as an actual parameter to the function called ‘b_search()’ to extract all the information regarding the book.



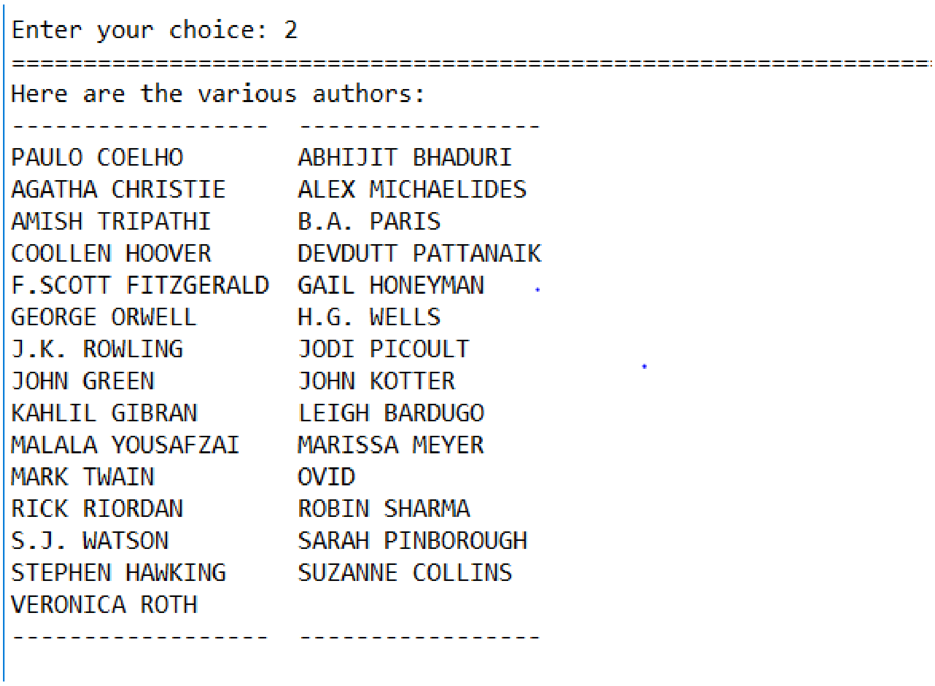
```
Enter your choice: 1
==========================================================================================================
Here are all the books in the library:
-----------------------------------------------------------  ---------------------------------------------
ABC MURDERS                                                  ALLEGIANT (DIVERGENT,#3)
AND THEN THERE WERE NONE                                     ANIMAL FARM
BEFORE I GO TO SLEEP                                         BEHIND CLOSED DOORS
BEHIND HER EYES                                              BREIF ANSWERS TO THE BIG QUESTIONS
BRIDA                                                        CATCHING FIRE (HUNGER GAMES,#2)
CINDER (THE LUNAR CHRONICLES,#1)                             CRESS (THE LUNAR CHRONICLES,#3)
CROOKED KINGDOM (SIX OF CROWS,#2)                            DIVERGENT (DIVERGENT,#1)
DON'T HIRE THE BEST                                          ELEANOR OLIPHANT IS COMPLETELY FINE
FAIREST (THE LUNAR CHRONICLES,#3.5)                          FAULT IN OUR STARS
FOUR (DIVERGENT,#4)                                          GREAT GATSBY
HARRY POTTER AND THE CHAMBER OF SECRETS (Harry Potter,#2)    HARRY POTTER AND THE DEATHLY HALLOWS (Harry Potter,#7)
HARRY POTTER AND THE GOBLET OF FIRE (Harry Potter,#4)        HARRY POTTER AND THE HALF-BLOOD PRINCE (Harry Potter,#6)
HARRY POTTER AND THE ORDER OF THE PHOENIX (Harry Potter,#5)  HARRY POTTER AND THE PHILOSOPHER'S STONE (Harry Potter,#1)
HARRY POTTER AND THE PRISONER OF AZKABAN (Harry Potter,#3)   I AM MALALA
INSURGENT (DIVERGENT,#2)                                     JAYA
LOOKING FOR ALASKA                                           METAMORPHOSIS
MOCKINGJAY (HUNGER GAMES,#3)                                 MURDER ON THE NILE
MY SISTERS KEEPER                                            OUR ICEBERGS ARE MELTING
PAPER TOWNS                                                  PERCY JACKSON AND THE BATTLE OF THE LABYRINTH (PERCY JACKSON,#4)
PERCY JACKSON AND THE LAST OLYMPIAN (PERCY JACKSON,#5)       PERCY JACKSON AND THE LIGHTNING THIEF (PERCY JACKSON,#1)
PERCY JACKSON AND THE SEA OF MONSTERS (PERCY JACKSON,#2)     PERCY JACKSON AND THE TITAN'S CURSE (PERCY JACKSON,#3)
RUINS AND RISING (THE GRISHA TRILOGY,#3)                     SCARLET (THE LUNAR CHRONICLES,#2)
SHADOW AND BONE (THE GRISHA TRILOGY,#1)                      SIEGE AND STORM (THE GRISHA TRILOGY,#2)
SITA                                                         SITA-THE WARRIOR OF MITHILA
SIX OF CROWS (SIX OF CROWS,#1)                               STARS ABOVE (THE LUNAR CHRONICLES,#4.5)
THE ADVENTURES OF TOM SAWYER                                 THE ALCHEMIST
THE HUNGER GAMES (HUNGER GAMES,#1)                           THE MONK WHO SOLD HIS FERRARI
THE MURDER OF ROGER ACKROYED                                 THE MURDER ON THE ORIENT EXPRESS
THE PROPHET                                                  THE SCION OF ISHKVAKU
THE SILENT PATIENT                                           THE THEORY OF EVERYTHING
THE TIME MACHINE                                             VERITY
VERONICA DECIDES TO DIE                                      WINTER (THE LUNAR CHRONICLES,#4)
-----------------------------------------------------------  ----------------------------------------------------------------
```



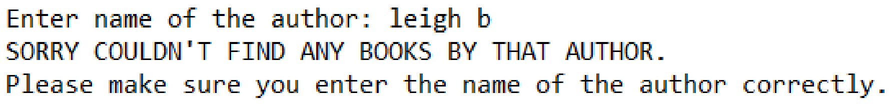
```
Enter the book name : six of crows
SORRY COULDN'T FIND THAT BOOK.
Please make sure you enter the correct book name.
```

```
elif int(ch)==2:
        print('='*138)
        cursor.execute("select distinct author from book order by author;")
        l=cursor.fetchall()
```

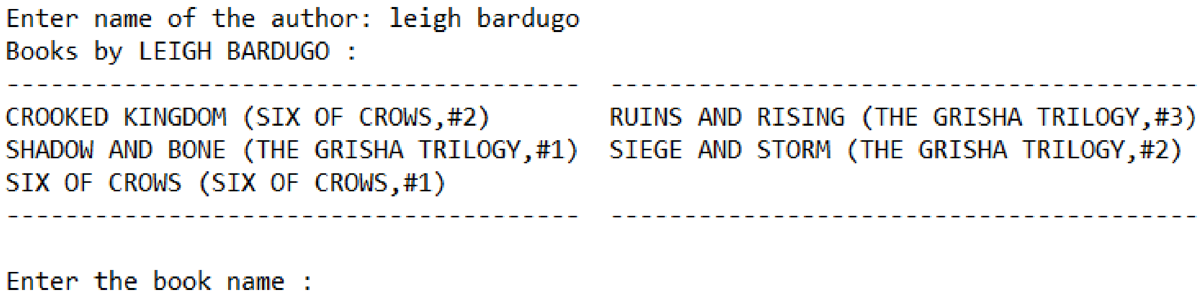
When the second option is selected it shows the various authors whose books are available in the library ascending order. The tabulate function allows us to display the data in an organised way. When the name of the author is entered, it checks whether name is a valid entry or not; if it is not then an appropriate message is displayed and the author name is asked to enter again. If the author name entered is correct then the loop breaks and the author name is sent as an actual parameter to the function called ‘a_search()’ to extract a list of various books by the given author in ascending order.



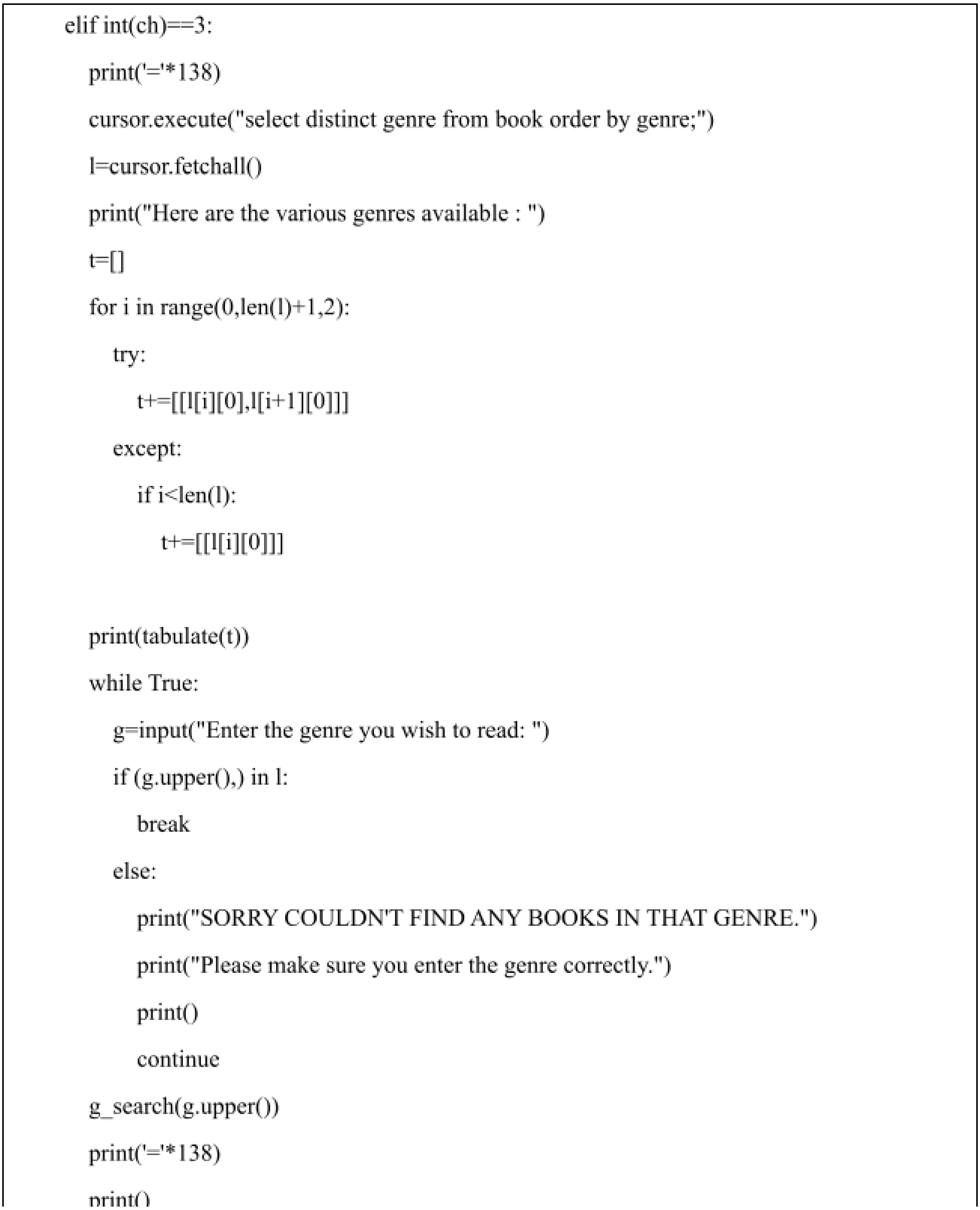
```
Enter your choice: 2
=================================================================
Here are the various authors:
------------------  -----------------
PAULO COELHO        ABHIJIT BHADURI
AGATHA CHRISTIE     ALEX MICHAELIDES
AMISH TRIPATHI      B.A. PARIS
COOLLEN HOOVER      DEVDUTT PATTANAIK
F.SCOTT FITZGERALD  GAIL HONEYMAN
GEORGE ORWELL       H.G. WELLS
J.K. ROWLING        JODI PICOULT
JOHN GREEN          JOHN KOTTER
KAHLIL GIBRAN       LEIGH BARDUGO
MALALA YOUSAFZAI    MARISSA MEYER
MARK TWAIN          OVID
RICK RIORDAN        ROBIN SHARMA
S.J. WATSON         SARAH PINBOROUGH
STEPHEN HAWKING     SUZANNE COLLINS
VERONICA ROTH
------------------  -----------------
```



```
Enter name of the author: leigh b
SORRY COULDN'T FIND ANY BOOKS BY THAT AUTHOR.
Please make sure you enter the name of the author correctly.
```

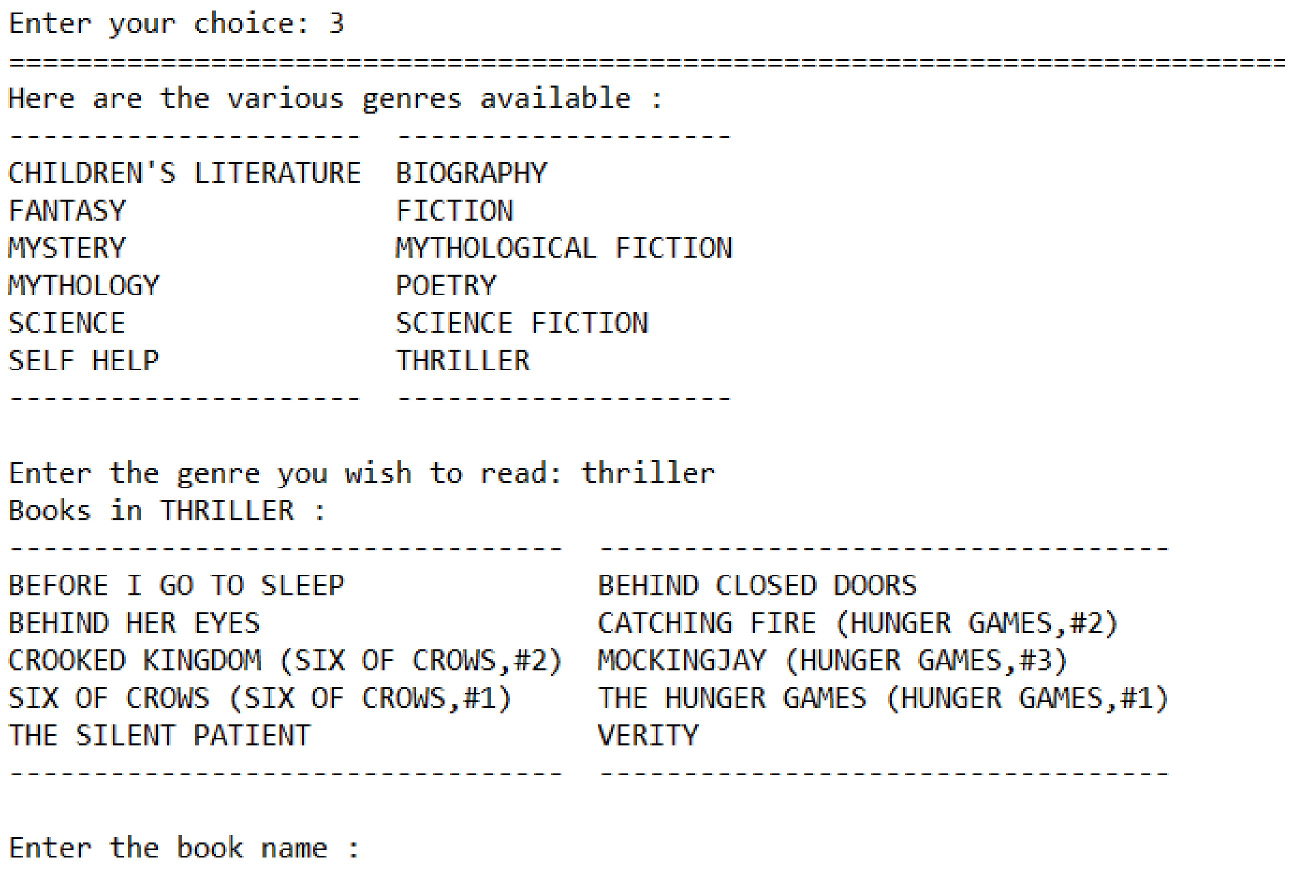


```
Enter name of the author: leigh bardugo
Books by LEIGH BARDUGO :
----------------------------------------  -----------------------------------------
CROOKED KINGDOM (SIX OF CROWS,#2)         RUINS AND RISING (THE GRISHA TRILOGY,#3)
SHADOW AND BONE (THE GRISHA TRILOGY,#1)   SIEGE AND STORM (THE GRISHA TRILOGY,#2)
SIX OF CROWS (SIX OF CROWS,#1)
----------------------------------------  -----------------------------------------

Enter the book name :
```

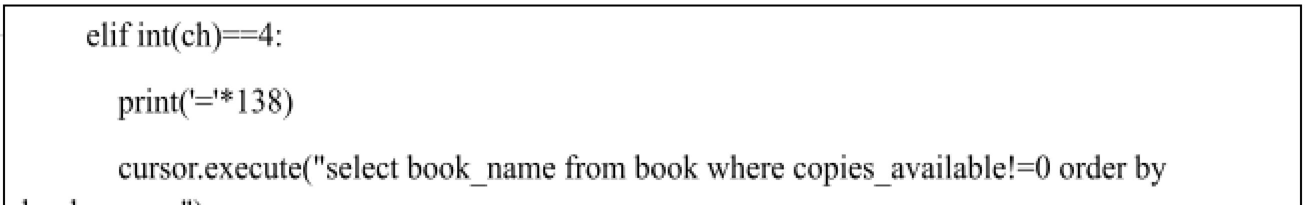


```
elif int(ch)==3:
    print('='*138)
    cursor.execute("select distinct genre from book order by genre;")
    l=cursor.fetchall()
    print("Here are the various genres available : ")
    t=[]
    for i in range(0,len(l)+1,2):
        try:
            t+=[[l[i][0],l[i+1][0]]]
        except:
            if i<len(l):
                t+=[[l[i][0]]]

    print(tabulate(t))
    while True:
        g=input("Enter the genre you wish to read: ")
        if (g.upper(),) in l:
            break
        else:
            print("SORRY COULDN'T FIND ANY BOOKS IN THAT GENRE.")
            print("Please make sure you enter the genre correctly.")
            print()
            continue
    g_search(g.upper())
    print('='*138)
    print()
```

When the third option is selected it shows the various books available in the library ascending order. The tabulate function allows us to display the data in an organised way. When the genre is entered, it checks whether it is a valid entry or not; if it is not then an appropriate message is displayed and the genre is asked to enter again. If the entered genre is correct then the loop breaks and the genre is sent as an actual parameter to the function called ‘g_search()’ to extract a list of various books in the given genre.

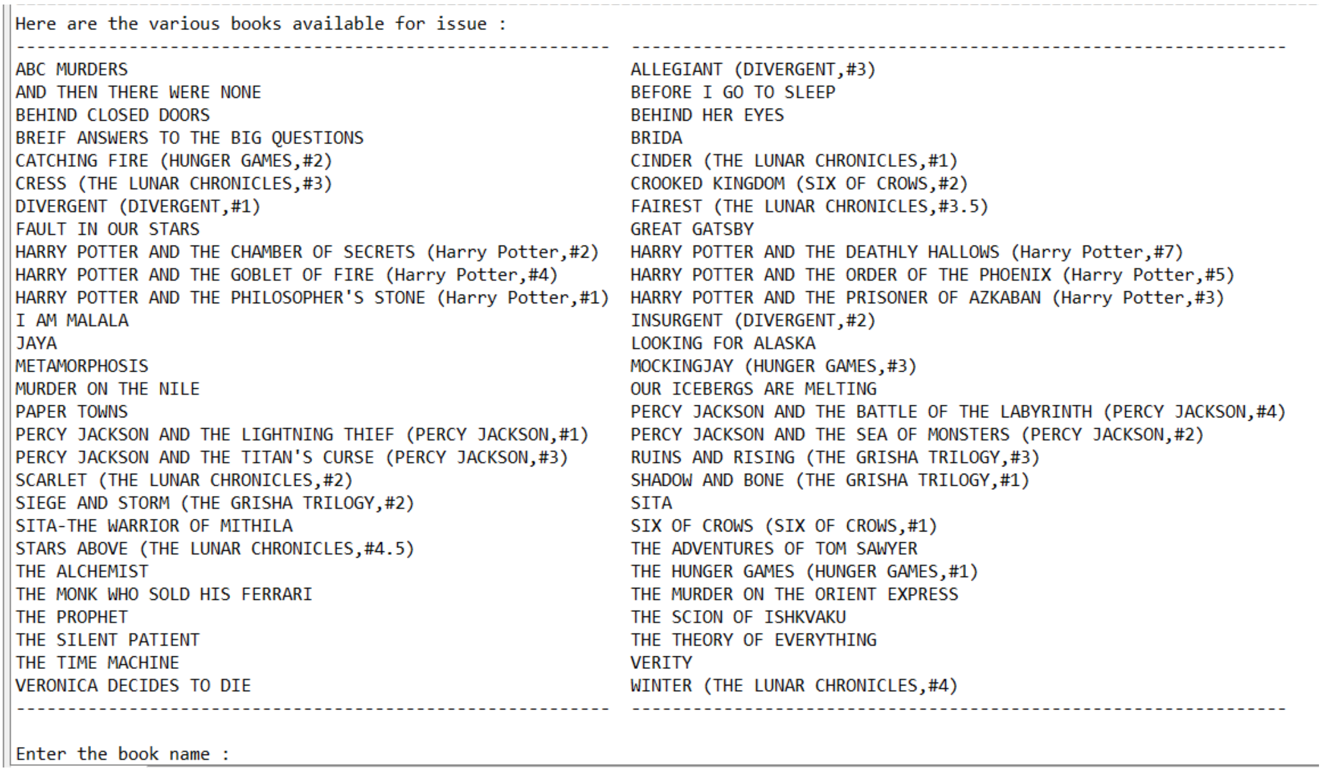


```
Enter your choice: 3
============================================================================
Here are the various genres available :
---------------------  --------------------
CHILDREN'S LITERATURE  BIOGRAPHY
FANTASY                FICTION
MYSTERY                MYTHOLOGICAL FICTION
MYTHOLOGY              POETRY
SCIENCE                SCIENCE FICTION
SELF HELP              THRILLER
---------------------  --------------------

Enter the genre you wish to read: thriller
Books in THRILLER :
---------------------------------  ----------------------------------
BEFORE I GO TO SLEEP               BEHIND CLOSED DOORS
BEHIND HER EYES                    CATCHING FIRE (HUNGER GAMES,#2)
CROOKED KINGDOM (SIX OF CROWS,#2)  MOCKINGJAY (HUNGER GAMES,#3)
SIX OF CROWS (SIX OF CROWS,#1)     THE HUNGER GAMES (HUNGER GAMES,#1)
THE SILENT PATIENT                 VERITY
---------------------------------  ----------------------------------

Enter the book name :
```



```
    elif int(ch)==4:
        print('='*138)
        cursor.execute("select book_name from book where copies_available!=0 order by
```

When the fourth option is selected it shows all the books that are available for issue ie the books whose number of copies I the library are zero are not displayed. The tabulate function allows us to display the data in an organised way.

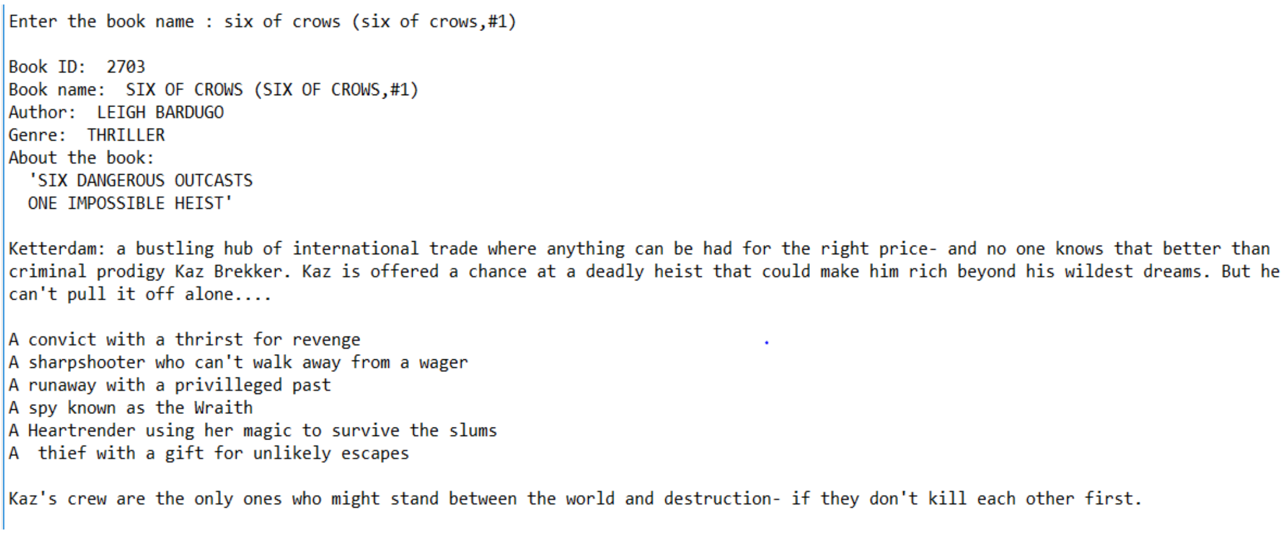


```
Here are the various books available for issue :
----------------------------------------------------------  ----------------------------------------------------------------
ABC MURDERS                                                 ALLEGIANT (DIVERGENT,#3)
AND THEN THERE WERE NONE                                    BEFORE I GO TO SLEEP
BEHIND CLOSED DOORS                                         BEHIND HER EYES
BREIF ANSWERS TO THE BIG QUESTIONS                          BRIDA
CATCHING FIRE (HUNGER GAMES,#2)                             CINDER (THE LUNAR CHRONICLES,#1)
CRESS (THE LUNAR CHRONICLES,#3)                             CROOKED KINGDOM (SIX OF CROWS,#2)
DIVERGENT (DIVERGENT,#1)                                    FAIREST (THE LUNAR CHRONICLES,#3.5)
FAULT IN OUR STARS                                          GREAT GATSBY
HARRY POTTER AND THE CHAMBER OF SECRETS (Harry Potter,#2)   HARRY POTTER AND THE DEATHLY HALLOWS (Harry Potter,#7)
HARRY POTTER AND THE GOBLET OF FIRE (Harry Potter,#4)       HARRY POTTER AND THE ORDER OF THE PHOENIX (Harry Potter,#5)
HARRY POTTER AND THE PHILOSOPHER'S STONE (Harry Potter,#1)  HARRY POTTER AND THE PRISONER OF AZKABAN (Harry Potter,#3)
I AM MALALA                                                 INSURGENT (DIVERGENT,#2)
JAYA                                                        LOOKING FOR ALASKA
METAMORPHOSIS                                               MOCKINGJAY (HUNGER GAMES,#3)
MURDER ON THE NILE                                          OUR ICEBERGS ARE MELTING
PAPER TOWNS                                                 PERCY JACKSON AND THE BATTLE OF THE LABYRINTH (PERCY JACKSON,#4)
PERCY JACKSON AND THE LIGHTNING THIEF (PERCY JACKSON,#1)    PERCY JACKSON AND THE SEA OF MONSTERS (PERCY JACKSON,#2)
PERCY JACKSON AND THE TITAN'S CURSE (PERCY JACKSON,#3)      RUINS AND RISING (THE GRISHA TRILOGY,#3)
SCARLET (THE LUNAR CHRONICLES,#2)                           SHADOW AND BONE (THE GRISHA TRILOGY,#1)
SIEGE AND STORM (THE GRISHA TRILOGY,#2)                     SITA
SITA-THE WARRIOR OF MITHILA                                 SIX OF CROWS (SIX OF CROWS,#1)
STARS ABOVE (THE LUNAR CHRONICLES,#4.5)                     THE ADVENTURES OF TOM SAWYER
THE ALCHEMIST                                               THE HUNGER GAMES (HUNGER GAMES,#1)
THE MONK WHO SOLD HIS FERRARI                               THE MURDER ON THE ORIENT EXPRESS
THE PROPHET                                                 THE SCION OF ISHKVAKU
THE SILENT PATIENT                                          THE THEORY OF EVERYTHING
THE TIME MACHINE                                            VERITY
VERONICA DECIDES TO DIE                                     WINTER (THE LUNAR CHRONICLES,#4)
----------------------------------------------------------  ----------------------------------------------------------------

Enter the book name :
```

b_search(book)

```
def b_search(book):
    cursor.execute("select book_id,book_name,author,genre,synopsis from book where
book_name=%s;",(book,))
```

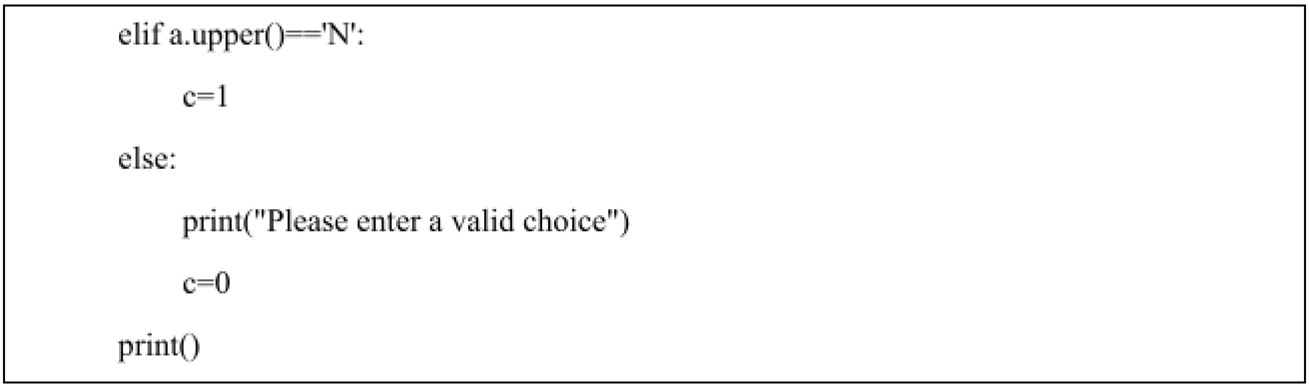
Once the book name entered by the member is accepted, it is sent as an actual parameter to the function b_search(book). Here the ‘book’ table is searched using a where condition. Once the data is fetched, it is displayed in a organised way as shown below.



```
Enter the book name : six of crows (six of crows,#1)

Book ID:  2703
Book name:  SIX OF CROWS (SIX OF CROWS,#1)
Author:  LEIGH BARDUGO
Genre:  THRILLER
About the book:
  'SIX DANGEROUS OUTCASTS
  ONE IMPOSSIBLE HEIST'

Ketterdam: a bustling hub of international trade where anything can be had for the right price- and no one knows that better than
criminal prodigy Kaz Brekker. Kaz is offered a chance at a deadly heist that could make him rich beyond his wildest dreams. But he
can't pull it off alone....

A convict with a thrirst for revenge
A sharpshooter who can't walk away from a wager
A runaway with a privilleged past
A spy known as the Wraith
A Heartrender using her magic to survive the slums
A  thief with a gift for unlikely escapes

Kaz's crew are the only ones who might stand between the world and destruction- if they don't kill each other first.
```

issue(book)

```
def issue(book):
    cursor.execute("select copies_available from book where book_name=%s;",(book,))
    l=cursor.fetchall()
    for i in l:
        if i==(0,):
            print()
            print("BOOK IS CURRENTLY UNAVAILABLE FOR ISSUE")
        else:
            c=0
            print()
            print("Do you wish to issue the book?")
            while c==0:
                a=input("Enter (Y/N): ")
                if a.upper()=='Y':
                    cursor.execute("select * from detail where member=%s and status='Issued';",(name,))
```



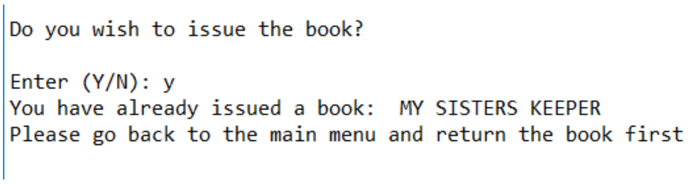
```
elif a.upper()=='N':
    c=1
else:
    print("Please enter a valid choice")
    c=0
print()
```

When the book name is entered by the member, it is sent as an actual parameter to the function issue(book). Firstly, the number of copies of the book available are fetched and if it is zero then an appropriate message is displayed stating that the book is not available for issue.

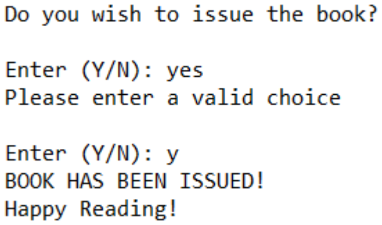
If copies are available then the program asks whether the member wishes to issue the said book. If yes , then firstly data is fetched from the table ‘details’ to check whether the member has returned the previously issued book or not by checking the status column in the table. If the member hasn’t returned the previous book an appropriate message is displayed, asking the member to return the book first.

If the member has returned the previous book then the said book is issued in the name of the member and the details of this issued book are stored in the details table for future use. The value of copies available is reduced by one in the books table.

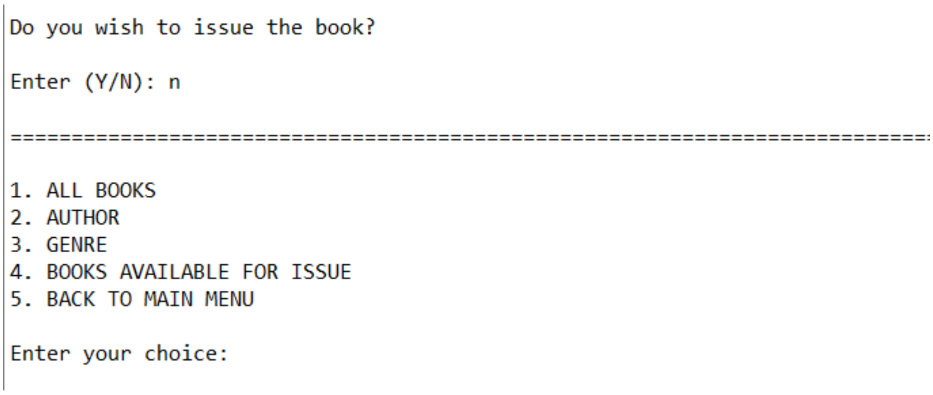
If the member says that he/she doesn’t wish to issue the book then the function terminates. If the member enters something different from what is asked then an appropriate message is displayed, asking to enter a valid entry.



```
Do you wish to issue the book?

Enter (Y/N): y
You have already issued a book:  MY SISTERS KEEPER
Please go back to the main menu and return the book first
```



```
Do you wish to issue the book?

Enter (Y/N): yes
Please enter a valid choice

Enter (Y/N): y
BOOK HAS BEEN ISSUED!
Happy Reading!
```



```
Do you wish to issue the book?

Enter (Y/N): n

===========================================================================

1. ALL BOOKS
2. AUTHOR
3. GENRE
4. BOOKS AVAILABLE FOR ISSUE
5. BACK TO MAIN MENU

Enter your choice:
```

return_b()

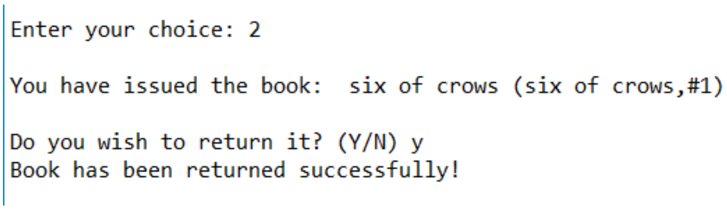
```
def return_b():
    cursor.execute("select * from detail where member=%s and status='Issued';",(name,))
    l=cursor.fetchall()
    print("You have issued the book: ",l[0][1])
    c=0
```

When the second option is selected in the menu(), the return_b() function is called. In this function data is fetched from the table detail by matching the member name and where the status is ‘Issued’. Then an appropriate message is displayed information the member what his/her last issued book was.

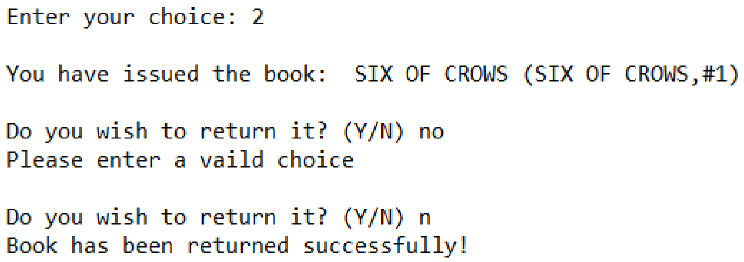
Then the member is asked if he/she wants to return the book. If yes, then the number of copies available is simply increased by 1 for the given book in the book table and a message is displayed stating that the book has been returned successfully.

If the member enters ‘N’ then we return back to the menu again.

If the member enters something else then the member is asked to enter a valid answer.



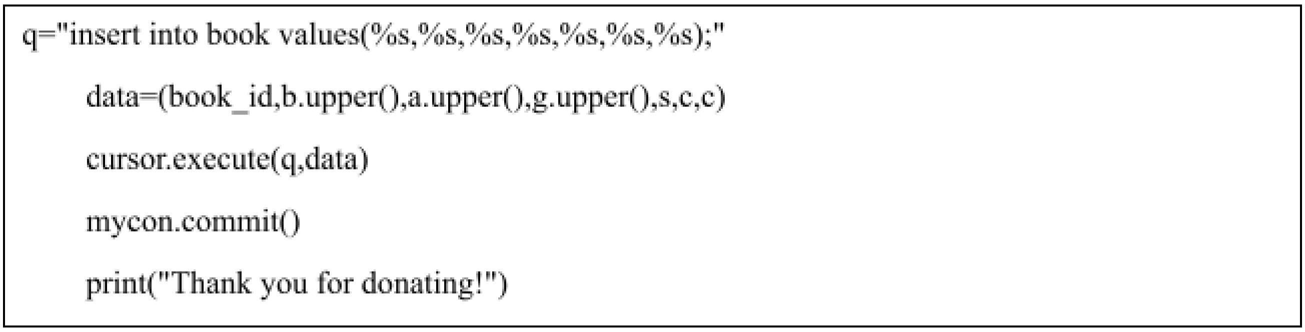
```
Enter your choice: 2

You have issued the book:  six of crows (six of crows,#1)

Do you wish to return it? (Y/N) y
Book has been returned successfully!
```



```
Enter your choice: 2

You have issued the book:  SIX OF CROWS (SIX OF CROWS,#1)

Do you wish to return it? (Y/N) no
Please enter a vaild choice

Do you wish to return it? (Y/N) n
Book has been returned successfully!
```

donate()

```
def donate():
    import random
    b=input("Enter the name of the book: ")
    a=input("Enter name of the author: ")
    g=input("Enter genre of the book: ")
    s=input("Synopsis(if any): ")
    c=int(input("Enter the number of copies you wish to donate: "))
    cursor.execute('select book_name from book;')
    l=cursor.fetchall()
    while (b.upper(),) in l:
        cursor.execute('update book set copies=copies+%s,copies_available=copies_available+%s where book_name=%s;',(c,c,b.upper()))
        mycon.commit()
        print("Thank you for donating!")
        break
    else:
        cursor.execute("select book_id from book;")
        l=cursor.fetchall()
        f=True
        while f:
            book_id=random.randint(1000,9999)
            for i in l:
                if i==(book_id,):
                    pass
                else:
                    f=False
```



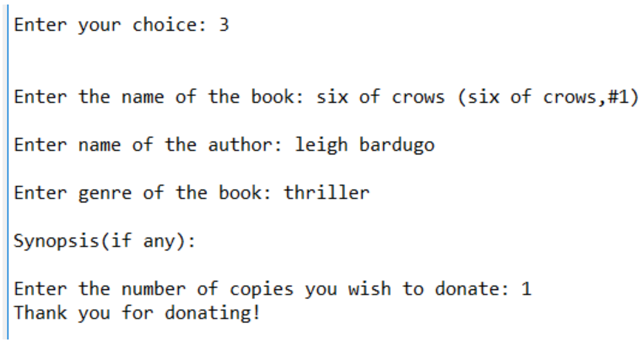
```
q="insert into book values(%s,%s,%s,%s,%s,%s,%s);"
        data=(book_id,b.upper(),a.upper(),g.upper(),s,c,c)
        cursor.execute(q,data)
        mycon.commit()
        print("Thank you for donating!")
```

When the third option is selected in the menu(), the donate() function is called.
If the book entered already exists in the book table, then the values of the copies_available column is simply increased by the number of copies donated.

Else a book id is generated randomly between 1000 and 9999
(both inclusive) using the randint() function of random library. Then this generated book id is checked with all the other book ids that already exist and if it is similar to one of them, then a new number is generated. Else the donated book is added to the book table with the book id generated.

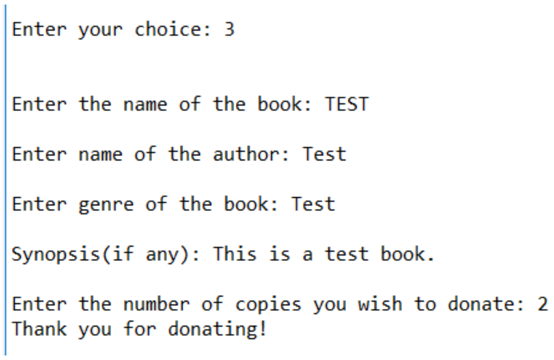
An appropriate message is displayed once the donation is successful.

For an already existing book :



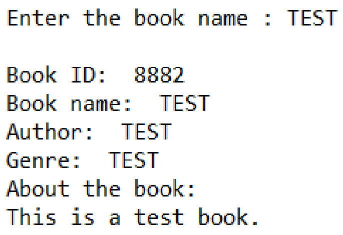
```
Enter your choice: 3


Enter the name of the book: six of crows (six of crows,#1)

Enter name of the author: leigh bardugo

Enter genre of the book: thriller

Synopsis(if any):

Enter the number of copies you wish to donate: 1
Thank you for donating!
```

For an different book that is not in the library



```
Enter your choice: 3


Enter the name of the book: TEST

Enter name of the author: Test

Enter genre of the book: Test

Synopsis(if any): This is a test book.

Enter the number of copies you wish to donate: 2
Thank you for donating!
```

As can be seen, a new book id has been generated and the data has been stored.



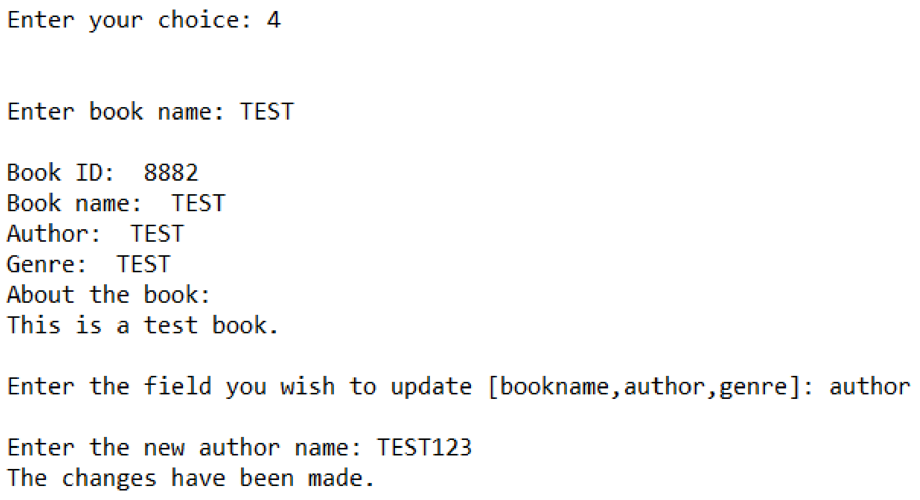
```
Enter the book name : TEST

Book ID:  8882
Book name:  TEST
Author:  TEST
Genre:  TEST
About the book:
This is a test book.
```

update()

```
def update():

    b=input("Enter book name: ")

    cursor.execute("select book_id,book_name,author,genre,synopsis from book where
elif f.upper()=='GENRE':
            g=input("Enter the new genre: ")
            cursor.execute("update book set genre=%s where book_name =%s;",(g.upper(),b.upper()))
            mycon.commit()
            h=1
        else:
            print("Please enter a valid choice: ")
    print("The changes have been made.")
    print()
```

This function allows user to update the information regarding any of the books. It first displays all the information about the book.

It then asks user to enter the field wherein the changes are to be made and according saves the changes in the table book.



```
Enter your choice: 4


Enter book name: TEST

Book ID:  8882
Book name:  TEST
Author:  TEST
Genre:  TEST
About the book:
This is a test book.

Enter the field you wish to update [bookname,author,genre]: author

Enter the new author name: TEST123
The changes have been made.
```

delete()

```
def delete():
    b=input("Enter the name of the book: ")
```

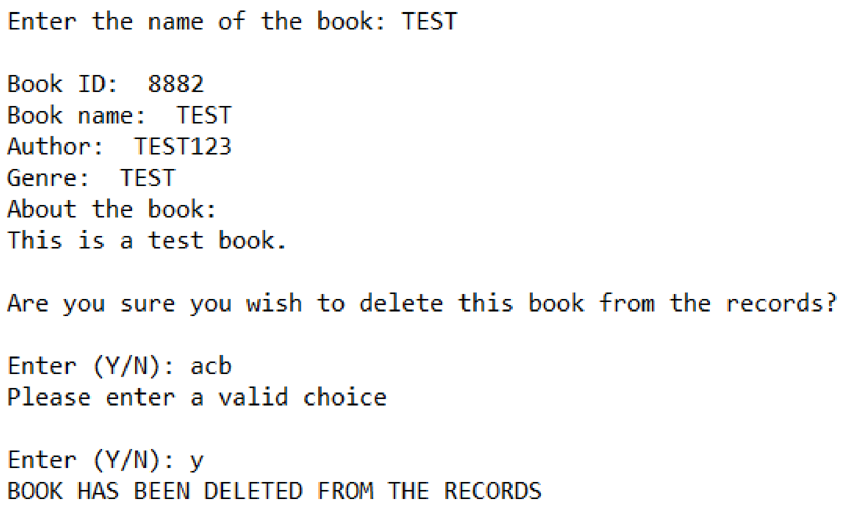
Suppose a book is published no more. The librarian can delete its record. Isn't it useful? This function does the same

This function allows the user a certain book from the table book. Firstly, the all the information about the book is displayed so that the user can confirm before deleting the data from the table.

After this, if the user enters yes then the data is deleted from the table and an appropriate message is displayed.

If the user enters no then the function is exited and the program continues.

If the user enters an invalid entry then an appropriate message is displayed and the user is asked to enter a valid choice.



```
Enter the name of the book: TEST

Book ID:  8882
Book name:  TEST
Author:  TEST123
Genre:  TEST
About the book:
This is a test book.

Are you sure you wish to delete this book from the records?

Enter (Y/N): acb
Please enter a valid choice

Enter (Y/N): y
BOOK HAS BEEN DELETED FROM THE RECORDS
```

DECLARATION

I , **RUJUTA PRAJAKT KULKARNI** , Roll No . a student of DPS , Navi Mumbai humbly submit that I have completed the project work as described in this report by my own skill and study as per the instructions of my teacher Ms. Madhurima Kashyap and that I have not copied the report or its any appreciable parts from any other literature in contravention of the academic ethics.

DATE : SIGNATURE OF THE STUDENT