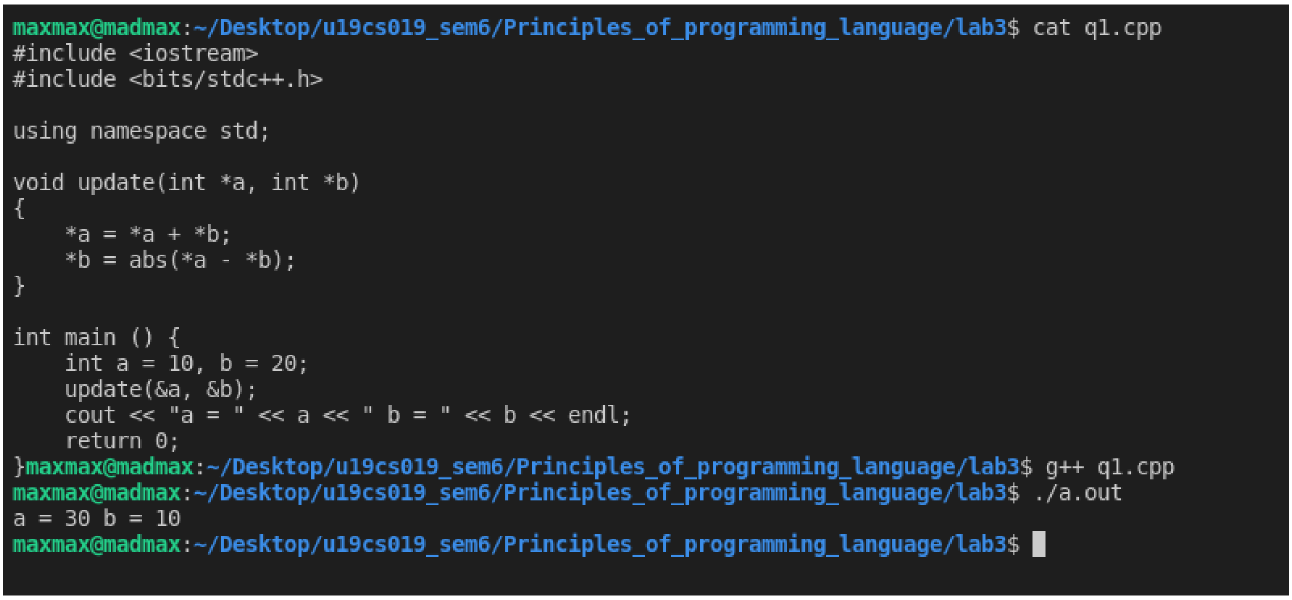
1. In order to access the memory address of a variable, val , prepend it with & sign. For example, &val returns the memory address of val. This memory address is assigned to a pointer and can be shared among functions. For example, int *p = &val assigns the memory address of val to pointer p. To access the content of the memory pointed to, prepend the variable name with a *. For example, *p will return the value stored in val and any modification to it will be performed on val. Create a program with function update having parameters as int *a & int *b Modify the values in memory so that a contains their sum and b contains their absolute difference.



```
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ cat q1.cpp
#include <iostream>
#include <bits/stdc++.h>

using namespace std;

void update(int *a, int *b)
{
    *a = *a + *b;
    *b = abs(*a - *b);
}

int main () {
    int a = 10, b = 20;
    update(&a, &b);
    cout << "a = " << a << " b = " << b << endl;
    return 0;
}maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ g++ q1.cpp
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ ./a.out
a = 30 b = 10
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ 
```

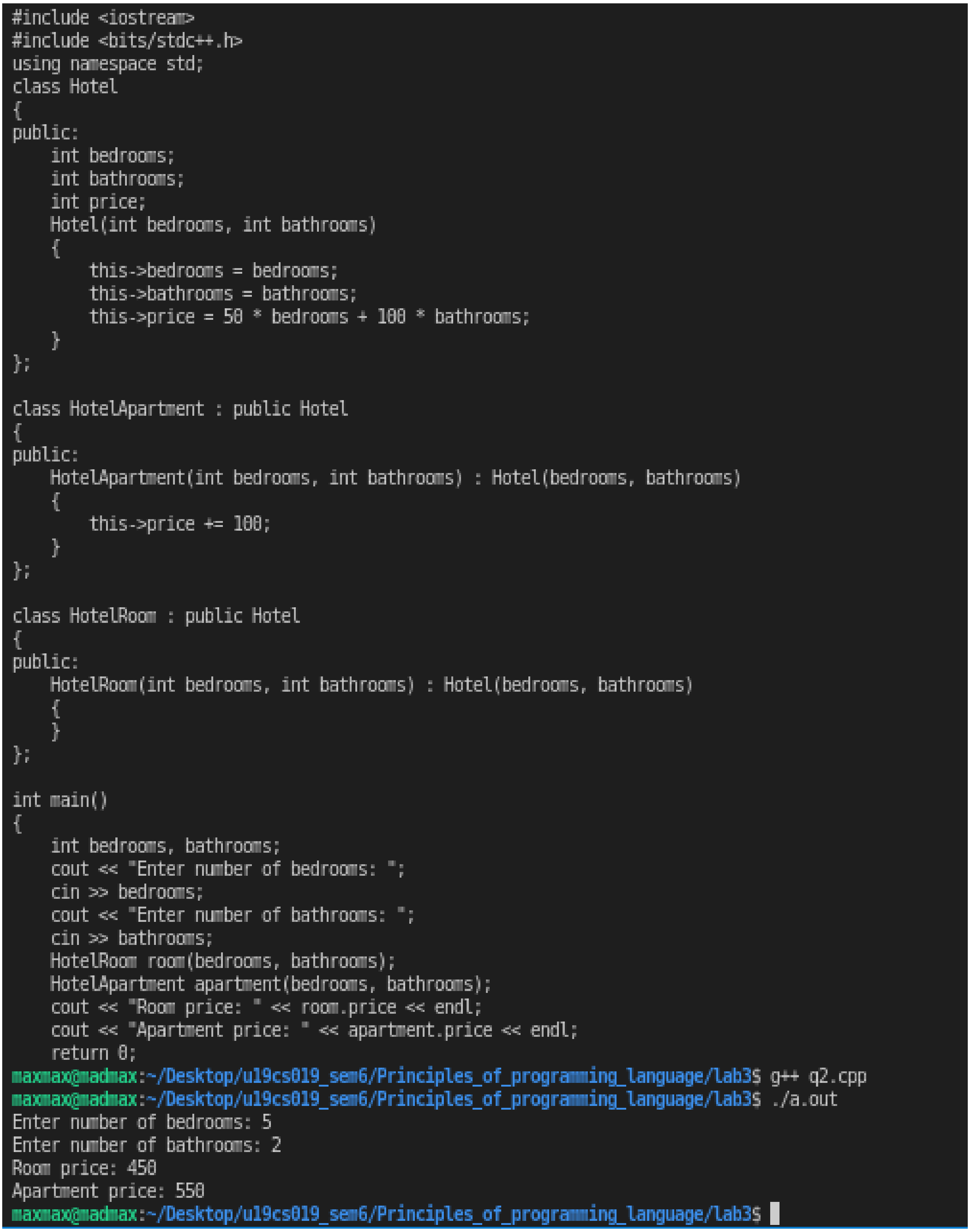
2. Write a program with two classes HotelRoom and HotelApartment denoting respectively a standard hotel room and a hotel apartment. An instance of any of these classes has two parameters: bedrooms and bathrooms denoting respectively the number of bedrooms and the number of bathrooms in the room.

The prices of a standard hotel room and hotel apartment are given as:

- Hotel Room: 50 x bedrooms + 100 x bathrooms.
- Hotel Apartment: The price of a standard room with the same number bedrooms and bathrooms plus 100.

For example, if a standard room costs 200, then an apartment with the same number of bedrooms and bathrooms costs 300.

Write a program to return the correct profit. Make necessary assumptions wherever necessary.



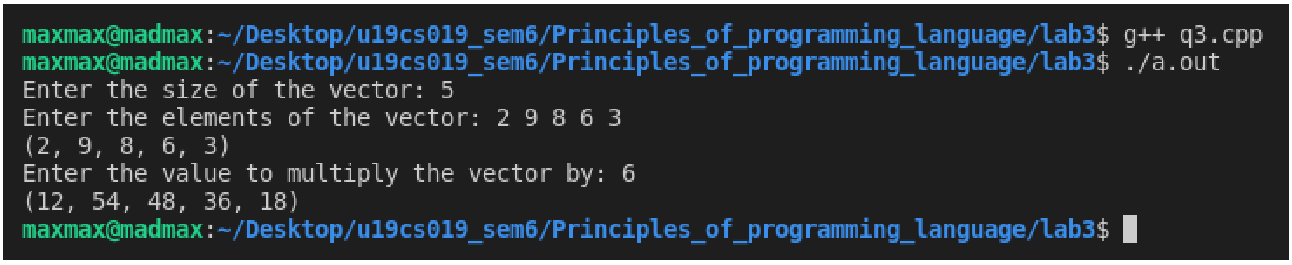
```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
class Hotel
{
public:
    int bedrooms;
    int bathrooms;
    int price;
    Hotel(int bedrooms, int bathrooms)
    {
        this->bedrooms = bedrooms;
        this->bathrooms = bathrooms;
        this->price = 50 * bedrooms + 100 * bathrooms;
    }
};

class HotelApartment : public Hotel
{
public:
    HotelApartment(int bedrooms, int bathrooms) : Hotel(bedrooms, bathrooms)
    {
        this->price += 100;
    }
};

class HotelRoom : public Hotel
{
public:
    HotelRoom(int bedrooms, int bathrooms) : Hotel(bedrooms, bathrooms)
    {
    }
};

int main()
{
    int bedrooms, bathrooms;
    cout << "Enter number of bedrooms: ";
    cin >> bedrooms;
    cout << "Enter number of bathrooms: ";
    cin >> bathrooms;
    HotelRoom room(bedrooms, bathrooms);
    HotelApartment apartment(bedrooms, bathrooms);
    cout << "Room price: " << room.price << endl;
    cout << "Apartment price: " << apartment.price << endl;
    return 0;
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ g++ q2.cpp
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ ./a.out
Enter number of bedrooms: 5
Enter number of bathrooms: 2
Room price: 450
Apartment price: 550
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ 
```

3. Write a class to represent a vector (a series of float values). Include member functions to perform the following tasks:

- To create the vector
- To modify the value of a given element.
- To multiply by a scalar value.
- To display the vector in the form (10, 20, 30,…)

```
#include <iostream>
#include <bits/stdc++.h>

using namespace std;

class Vector
{
public:
int size;
float *arr;
Vector(int size)
{
this->size = size;
arr = new float[size];
}
void set(int index, float value)
{
arr[index] = value;
}
void display()
{
cout << "(";
for(int i = 0; i < size; i++)
{
cout << arr[i];
if(i != size - 1)
cout << ", ";
}
cout << ")";
}
void multiply(float value)
{
for(int i = 0; i < size; i++)
arr[i] *= value;
}
};

int main () {
```

```
int size;
cout << "Enter the size of the vector: ";
cin >> size;
Vector v(size);
cout << "Enter the elements of the vector: ";
for(int i = 0; i < size; i++)
{
cin >> v.arr[i];
}
v.display();
cout << endl;
float value;
cout << "Enter the value to multiply the vector by: ";
cin >> value;
v.multiply(value);
v.display();
cout << endl;
return 0;
}
```



```
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ g++ q3.cpp
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ ./a.out
Enter the size of the vector: 5
Enter the elements of the vector: 2 9 8 6 3
(2, 9, 8, 6, 3)
Enter the value to multiply the vector by: 6
(12, 54, 48, 36, 18)
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$
```

4. A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise the message “Required copies not in stock” is displayed. Design a system using a class called books with suitable member functions and Constructors. Use new operators in constructors to allocate memory space required. Implement C++ program for the system.

Improve the system design to incorporate the following features:

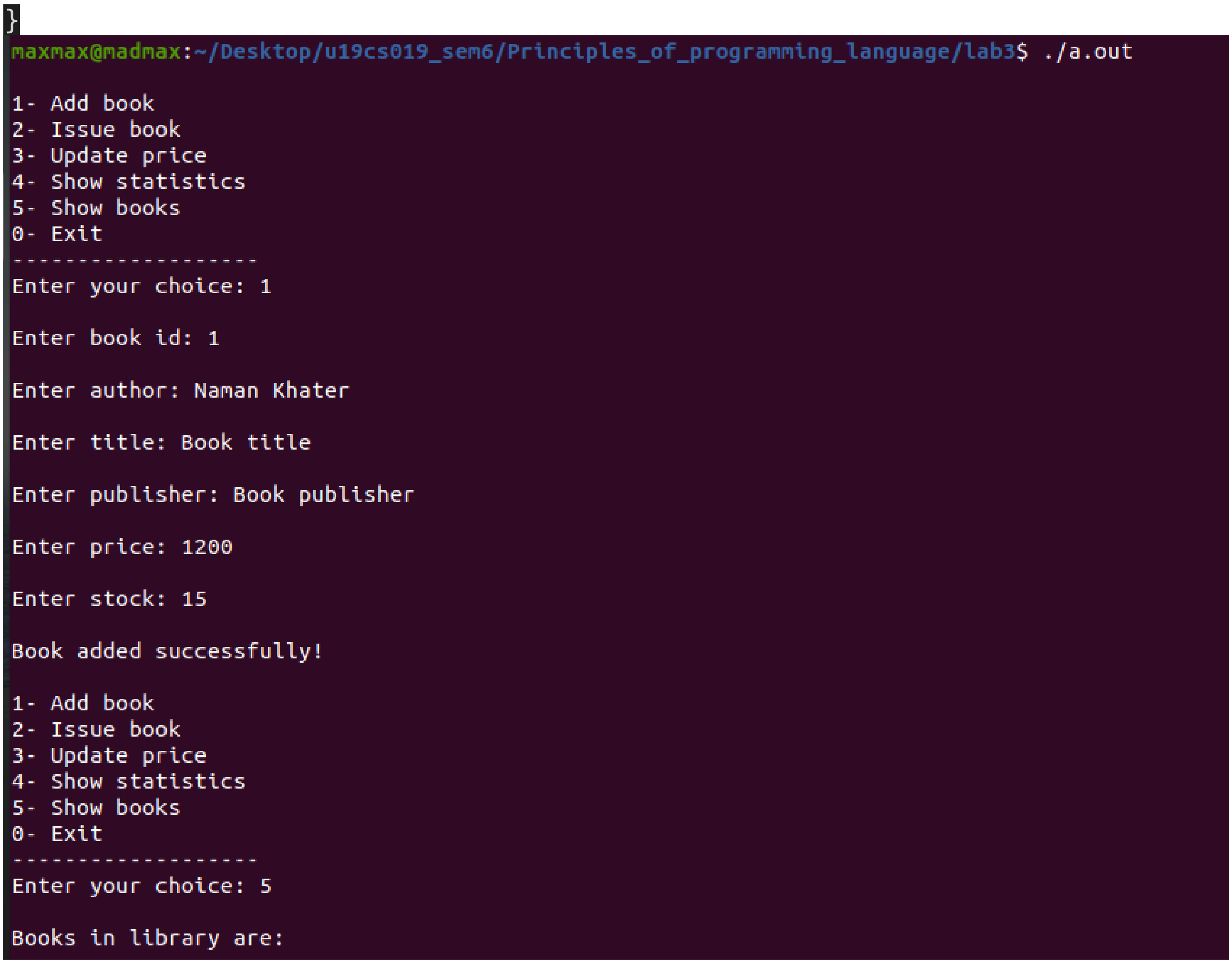
- The price of the books should be updated as and when required. Use a member function to implement this.
- The stock value of each book should be automatically updated as soon as a transaction is completed.

- The number of successful and unsuccessful transactions should be recorded for the purpose of statistical analysis. Use static data members to keep count of transactions.
- Also demonstrate the use of pointers to access the members.

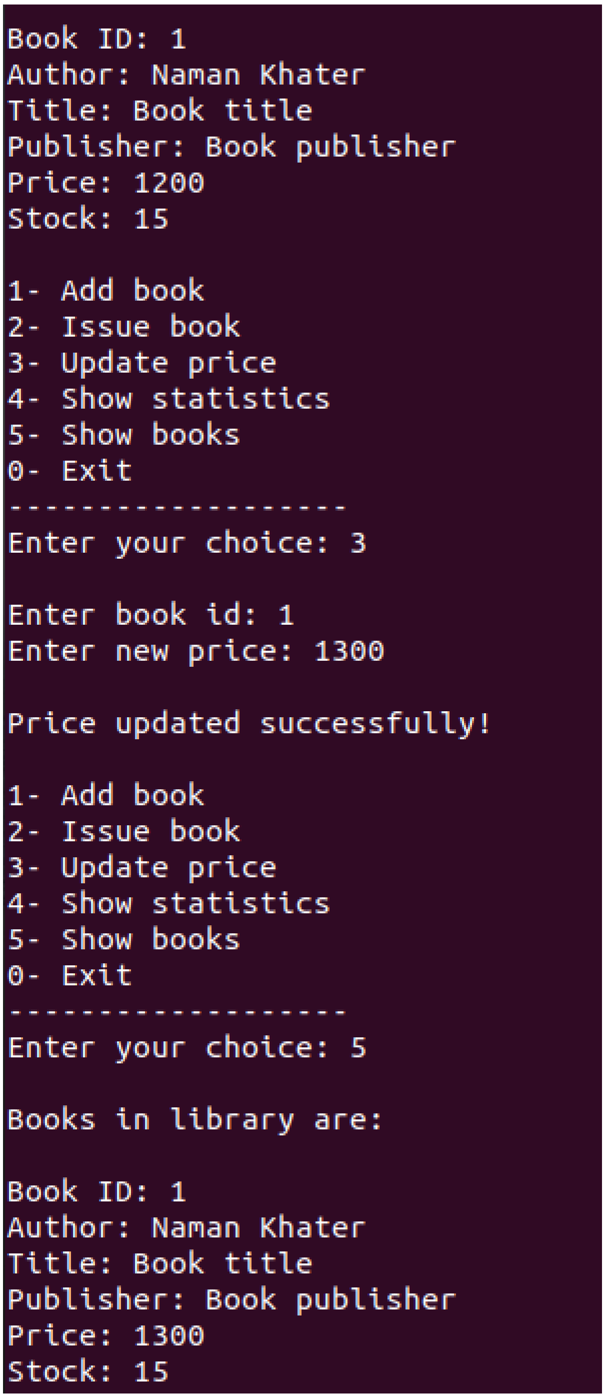
```
#include <bits/stdc++.h>
#include <limits>
#include <iostream>
using namespace std;
class Book
{
public:
int id, stock;
string author, title, publisher;
double price;
Book() {}
Book(int id, string author, string title, string publisher, double price,
int stock)
{
this->id = id;
this->author = author;
this->title = title;
this->publisher = publisher;
this->price = price;
this->stock = stock;
}
void updatePrice(double price)
{
this->price = price;
cout << "\nPrice updated successfully!" << endl;
}
};
class Library
{
public:
Book books[100];
int idx;
int successful, unsuccessful;
Library()
{
idx = 0;
successful = 0;
unsuccessful = 0;
}
void addBook(int id, string author, string title, string publisher, double price, int stock)
{
if (idx == 99)
{
cout << "\nLibrary is full" << endl;
return;
}
books[idx++] = Book(id, author, title, publisher, price, stock);
cout << "\nBook added successfully!" << endl;
}
void showBooks()
{
cout << "\nBooks in library are: " << endl;
for (int i = 0; i < idx; i++)
{
cout << "\nBook ID: " << books[i].id << endl;
cout << "Author: " << books[i].author << endl;
cout << "Title: " << books[i].title << endl;
cout << "Publisher: " << books[i].publisher << endl;
cout << "Price: " << books[i].price << endl;
cout << "Stock: " << books[i].stock << endl;
}
}
void issueBook(int id, int n)
{
for (int i = 0; i < idx; i++)
{
if (books[i].id == id)
{
if (books[i].stock < n)
{
cout << "\nNot enough books in stock. Only " << books[i].stock << " left." << endl;
unsuccessful++;
return;
}
books[i].stock -= n;
successful++;
cout << "\nBook issued successfully!" << endl;
return;
}
}
unsuccessful++;
cout << "\nBook not found" << endl;
}
};
int main()
{
Library library = Library();
int choice;
```

```
do
{
cout << "\n1- Add book" << endl;
cout << "2- Issue book" << endl;
cout << "3- Update price" << endl;
cout << "4- Show statistics" << endl;
cout << "5- Show books" << endl;
cout << "0- Exit" << endl;
cout << "-------------------" << endl;
cout << "Enter your choice: ";
cin >> choice;
switch (choice)
{
case 1:
{
int id, stock;
string author, title, publisher;
double price;
cout << "\nEnter book id: ";
cin >> id;
cout << "\nEnter author: ";
fflush(stdin);
getline(cin, author);
getline(cin, author);
cout << "\nEnter title: ";
fflush(stdin);
getline(cin, title);
cout << "\nEnter publisher: ";
fflush(stdin);
getline(cin, publisher);
cout << "\nEnter price: ";
fflush(stdin);
cin >> price;
cout << "\nEnter stock: ";
cin >> stock;
library.addBook(id, author, title, publisher, price, stock);
break;
}
case 2:
{
int n, id;
cout << "\nEnter book id: ";
cin >> id;
cout << "\nEnter number of books: ";
cin >> n;
library.issueBook(id, n);
break;
}
case 3:
{
double price;
int id;
cout << "\nEnter book id: ";
cin >> id;
cout << "Enter new price: ";
cin >> price;
bool found = false;
for (int i = 0; i < library.idx; i++)
{
if (library.books[i].id == id)
{
library.books[i].updatePrice(price);
library.successful++;
found = true;
break;
}
}
if (!found)
{
library.unsuccessful++;
cout << "\nBook not found" << endl;
}
break;
}
case 4:
{
cout << "\nSuccessful transactions: " << library.successful << endl;
cout << "Unsuccessful transactions: " << library.unsuccessful << endl;
break;
}
case 5:
{
library.showBooks();
break;
}
case 0:
{
cout << "\nThank you!" << endl;
break;
}
default:
cout << "\nInvalid choice" << endl;
```

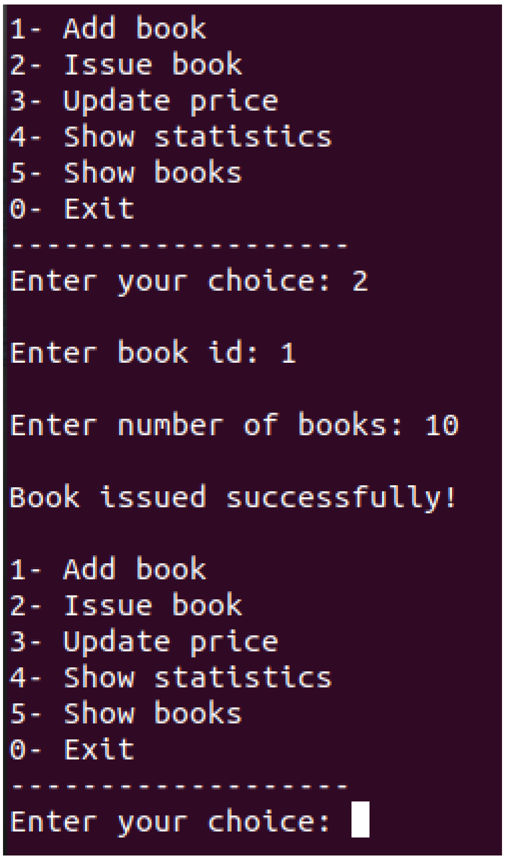
```
break;
}
} while (choice != 0);
return 0;
}
```



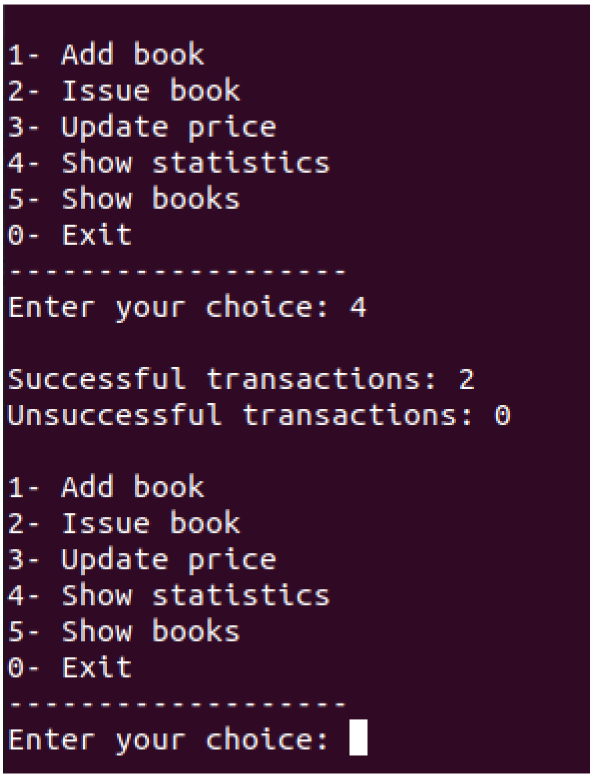
```
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ ./a.out

1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 1

Enter book id: 1

Enter author: Naman Khater

Enter title: Book title

Enter publisher: Book publisher

Enter price: 1200

Enter stock: 15

Book added successfully!

1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 5

Books in library are:
```



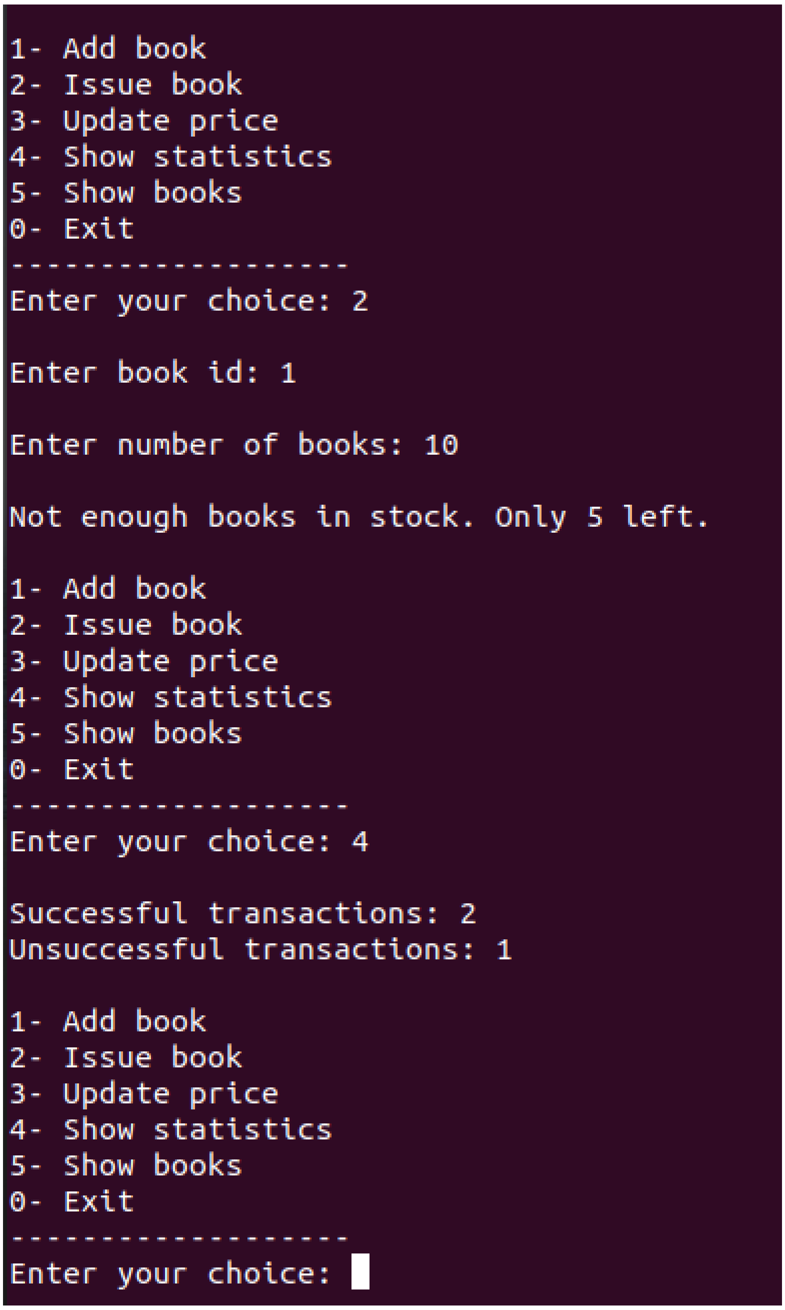
```
Book ID: 1
Author: Naman Khater
Title: Book title
Publisher: Book publisher
Price: 1200
Stock: 15

1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 3

Enter book id: 1
Enter new price: 1300

Price updated successfully!

1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 5

Books in library are:

Book ID: 1
Author: Naman Khater
Title: Book title
Publisher: Book publisher
Price: 1300
Stock: 15
```



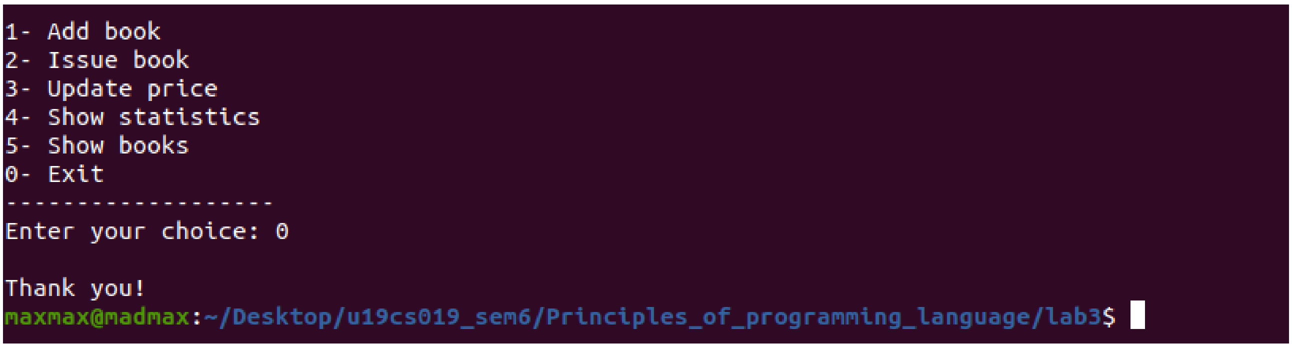
```
1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 2

Enter book id: 1

Enter number of books: 10

Book issued successfully!

1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 
```



```
1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 4

Successful transactions: 2
Unsuccessful transactions: 0

1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 
```



```
1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 2

Enter book id: 1

Enter number of books: 10

Not enough books in stock. Only 5 left.

1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 4

Successful transactions: 2
Unsuccessful transactions: 1

1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 
```



```
1- Add book
2- Issue book
3- Update price
4- Show statistics
5- Show books
0- Exit
-------------------
Enter your choice: 0

Thank you!
maxmax@madmax:~/Desktop/u19cs019_sem6/Principles_of_programming_language/lab3$ 
```