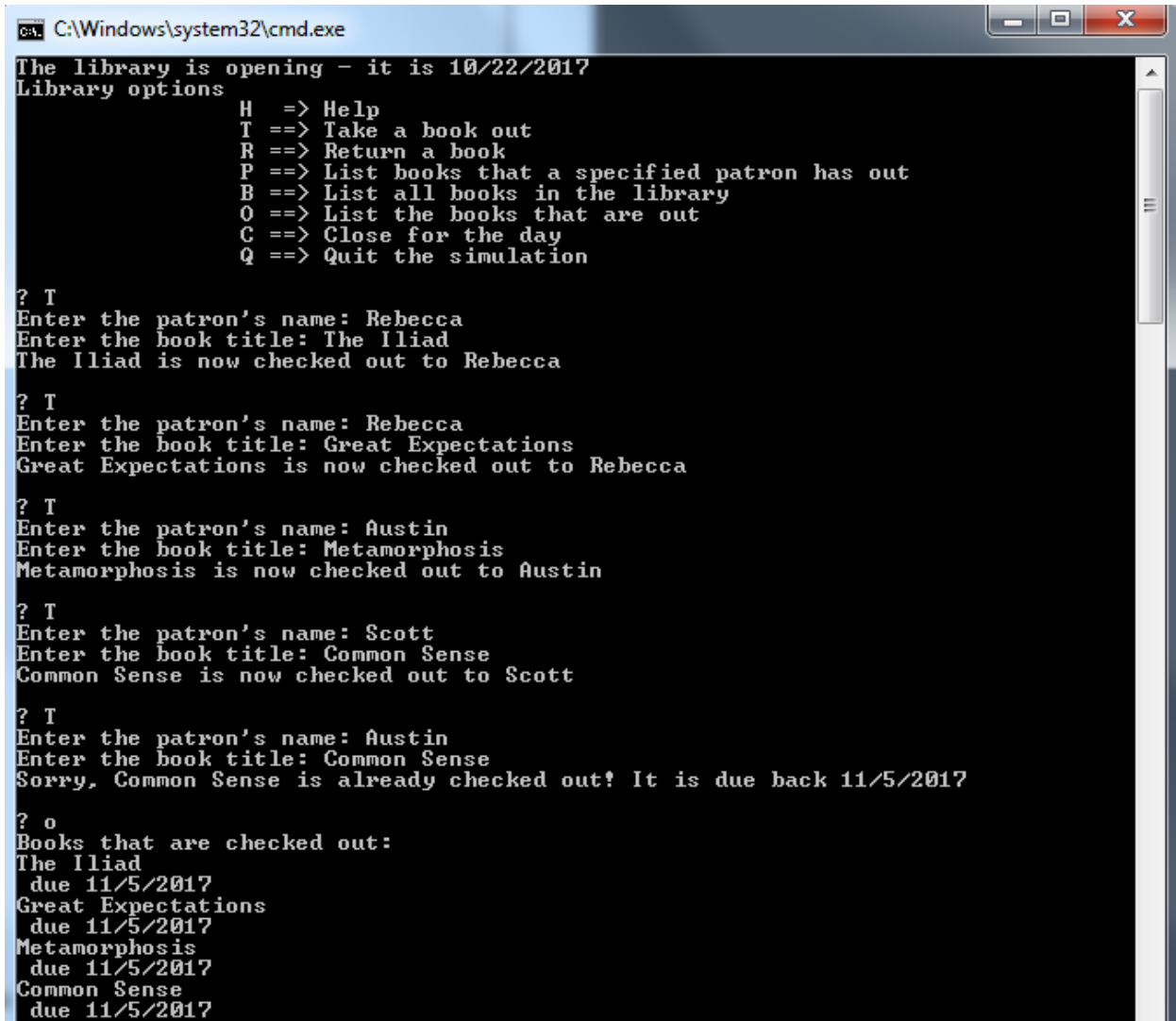


Lab 6 – Binary Search Tree

1. Screenshots of the program:

- Take a book out (T) and List the books that are out (O)



```
C:\Windows\system32\cmd.exe
The library is opening - it is 10/22/2017
Library options
      H ==> Help
      T ==> Take a book out
      R ==> Return a book
      P ==> List books that a specified patron has out
      B ==> List all books in the library
      O ==> List the books that are out
      C ==> Close for the day
      Q ==> Quit the simulation

? T
Enter the patron's name: Rebecca
Enter the book title: The Iliad
The Iliad is now checked out to Rebecca

? T
Enter the patron's name: Rebecca
Enter the book title: Great Expectations
Great Expectations is now checked out to Rebecca

? T
Enter the patron's name: Austin
Enter the book title: Metamorphosis
Metamorphosis is now checked out to Austin

? T
Enter the patron's name: Scott
Enter the book title: Common Sense
Common Sense is now checked out to Scott

? T
Enter the patron's name: Austin
Enter the book title: Common Sense
Sorry, Common Sense is already checked out! It is due back 11/5/2017

? o
Books that are checked out:
The Iliad
  due 11/5/2017
Great Expectations
  due 11/5/2017
Metamorphosis
  due 11/5/2017
Common Sense
  due 11/5/2017
```

- Return a book (R) and List books that a specified patron has out (P)

```
? t
Enter the patron's name: Rebecca
Enter the book title: The Iliad
The Iliad is now checked out to Rebecca

? T
Enter the patron's name: Rebecca
Enter the book title: Great Expectations
Great Expectations is now checked out to Rebecca

? p
Enter the patron's name: Rebecca
Books checked out by Rebecca:
The Iliad
  due 11/5/2017
Great Expectations
  due 11/5/2017

? r
Enter the book title: The Iliad
The Iliad
  has been checked in

? p
Enter the patron's name: Rebecca
Books checked out by Rebecca:
Great Expectations
  due 11/5/2017

? o
Books that are checked out:
Great Expectations
  due 11/5/2017
```

- Help (H), Close for the day (C) and Quit the simulation (Q)

```
C:\Windows\system32\cmd.exe
The library is opening - it is 10/22/2017
Library options
      H ==> Help
      T ==> Take a book out
      R ==> Return a book
      P ==> List books that a specified patron has out
      B ==> List all books in the library
      O ==> List the books that are out
      C ==> Close for the day
      Q ==> Quit the simulation

?
h
Library options
      H ==> Help
      T ==> Take a book out
      R ==> Return a book
      P ==> List books that a specified patron has out
      B ==> List all books in the library
      O ==> List the books that are out
      C ==> Close for the day
      Q ==> Quit the simulation

? c
Closing for the evening....
Opening again on 10/23/2017

? c
Closing for the evening....
Opening again on 10/24/2017

? q
Ending the simulation
Press any key to continue . . . _
```

2. Binary search tree:

- We create two binary search tree, one for patron name and one for book name.

```
BST<patron>patrontree;  
BST<book> booktree;
```

- Using these trees, we are able to store all the patron names as well as all the book names from the texts:

```
while (getline(file, line))  
{  
    patron p = patron(line);  
    patrontree.insert(p);  
}  
  
while (getline(file, line))  
{  
    string str = " by";  
    char *cstr = &str[0u];  
    size_t prev = 0, pos;  
    pos = line.find(str, prev);  
    if (pos > prev){  
        booktree.insert(line.substr(prev, pos - prev));  
    }  
}
```

- Binary search trees are used to check whether the patron name or the book name is in the list or not (using search function).

```
if (patrontree.search(tmp) == nullptr){  
    cout << "The patron name is not in the list.";  
    break;  
}  
  
if (booktree.search(tmpb) == nullptr){  
    cout << "The book name is not in the list.";  
    break;  
}
```

- In addition, they help with checking out/in a book in the library:

```
// check out  
if (booktree.search(tmpb)->borrow(patrontree.search(tmp), todaysDate)){  
    cout << bookname << " is now checked out  
to " << name << endl;  
  
    book b = *booktree.search(tmpb);  
    patrontree.search(tmp)->borrowBook(b);  
    bool st = false;  
    for (auto v : borrower){  
        if (tmp == v)  
            st = true;  
    }  
    if (!st)  
        borrower.push_back(tmp);  
}  
else
```

```

        cout << "Sorry, " << bookname << " is already
checked out! It is due back " << booktree.search(tmpb)->getdue() << endl;
//check in
if (booktree.search(tmpb) == nullptr){
    cout << "The book name is not in the list.";
    break;
}
else
{
    book b = *booktree.search(tmpb);
    patron *p = b.checkIn();
    if(p->returnBook(tmpb))
        cout << tmpb << " has been checked in " << endl;
}

```

- And two trees help us to output all books checked out by a specific patron as well as output all books checked out.

```

cout << "Books checked out by " << name << ": " << endl;
        patrontree.search(tmpp)->showBooks();

cout << "Books that are checked out: " << endl;
    for (auto p : borrower)
        patrontree.search(p)->showBooks();

```