Programming Assignment #5 - Binary Trees and Recursion (50 pts)

This assignment is similar to #4 except

- The Book information should be represented as an ascending ordered binary tree. (Keep the tree nodes in order as they are inserted.)
- Your search for a Book Id is using a recursive binary tree traversal.
- The command file will have a new subcommand, PPRINT, for BOOK. We will be able to pretty print the binary tree of books. See the output below.
- You will be provided with a driver program (cs1713p5Driver.c) (see below)
- Your code must be created in a separate C file (p5abc123.c). (see below)
- There is a new include file, cs1713p5.h
- Several of the functions must be recursive: searchT, insertT, prettyPrintT, printInorder
- We have provided a Makefile to reduce the chances of errors typing in your gcc commands.

Input:

Book — same as Programming Assignment #4; however, instead of placing it in linked list, you will put it into an ascending ordered binary tree. Some of the data may have changed.

Command — Same as assignment 4 plus this new subcommand for BOOK:

`BOOK PPRINT`

This pretty prints the binary tree. You only have to print the Book IDs in a pretty print manner. In the following example, notice that the right most child is printed first.

```
            TECHDR001
         SQLDBB001
            PYTHON002
      PYTHON001
            PRANKS001
         PERLLL001
            LINUXX004
   JOYPGM004
            JOYPGM003
         JOYPGM002
            JOYPGM001
      JAVADD001
            EXCELL001
         COBOLL001
            ARTINT001
```

Driver program:

You will be provided with a driver program, cs1713p5Driver.c which

1. invokes the driver's processCommandSwitches
2. invokes the driver's getBooks to read the original book information into a binary tree ordered by Book ID using your insertT function.
3. invokes your printBooks to print the original book information.
4. invokes a driver-provided processCommands which
    - reads input lines from the command file until eof:
        - prints the input line
        - determines command and subcommand
        - invokes either
            - your processCustomerCommand to process a CUSTOMER subcommand
            - your processBookCommand to process a BOOK subcommand
5. invokes your printBooks to print the resulting book information
6. Larry also provided these functions:

- allocateNodeT
- getBooks

Note: do **not change** the cs1713p5Driver.c

Your p5abc123.c code:

- You should probably copy your **p4abc123.c** into a file named **p5abc123.c**.
- It does the following includes:

```
#include <stdio.h>
#include <string.h>
#include "cs1713p5.h"
```

- It must **not** include `cs1713p5Driver.c` within your **p5abc123.c** file. Look at the notes below on **Compiling Using the make Utility.**
- Remove insertLL. We will be using insertT.
- Remove searchLL. We will be using searchT.
- Change printBooks:
  - Receives `NodeT *pRoot` instead of the linked list pHead.
  - It should still print the column heading for the table of books; however, it should invoke printInOrder to print the tree in order.
- Add the recursive function **printInOrder** which prints all the information about books in order recursively. This is called by printBooks.
- Change **processCustomerCommand**:
  - Receives NodeT *pRoot instead of the Node **ppHead. Notice that we are **not** passing the address of pRoot to this function.
  - Invokes processTransaction passing pRoot:

```
pCustomer->dFeeBalance += processTransaction(pRoot, customer, transaction);
```

- Change **processTransaction**:
  - Receives NodeT *pRoot instead of the Node *pHead
  - Uses searchT to find a book in the binary tree.
- Change **processBookCommand**:
  - Receives `NodeT **ppRoot` instead of the `Node **ppHead`.
  - Uses searchT to find a book in the binary tree. You will have to dereference ppRoot.
  - The NEW subcommand uses insertT to insert a new book into the binary tree.

```
*ppRoot = insertT(*ppRoot, book);
```

  - Add code for the new BOOK PPRINT subcommand. This should invoke prettyPrintT.
- Add the function **prettyPrintT** which prints a binary tree by printing its right most node first. You only have to print the book IDs.
- You must create the following routines (see the include file):
  - **insertT** - using the reconstruct approach, this recursively inserts a book into the ordered binary tree. This returns a pointer to the referenced subtree which is either the pointer it was passed or a pointer to a new node.
  - **searchT** - recursively searches for a Book Id in the ordered binary tree. If found, it returns a pointer to the node that contains it. If not found, it returns NULL.

Please review the cs1713p5.h include file.

Compiling Using the make Utility

Before using the make utility, you must:

- Download the Makefile.txt file and rename it simply **Makefile** (i.e., remove the .txt suffix)
- Edit Makefile to change `p5abc123.o` with your abc123 id. Note that you are changing a **.o** file reference. The make utility will automatically reference your .c file if you properly name the p5abc123.o file:

```
# Define the machine object files for your program
OBJECTS = p5abc123.o cs1713p5Driver.o
# Define your include file
```

```
INCLUDES = cs1713p5.h

# make for the executable
p5: ${OBJECTS}
      gcc -g -o p5 ${OBJECTS}

# Simple suffix rules for the .o
%.o: %.c ${INCLUDES}
      gcc -g -c $<

# Clean the .o files
clean:
      rm -f ${OBJECTS}
```

Based on the rules in the Makefile, when you tell make to make your executable, it will **automatically compile** your .c file and the driver .c file. (For more information about the **make** utility, click here.)

```
make p5
```

Executing the p5 executable:

```
./p5 -c p5Command.txt -b p5Book.txt
```

Turn in:

- Your include file (if it changed)
- Your p5abc123.c file.
- Your Makefile (since you changed it for your p5abc123.c). The TA/grader will use your Makefile to make the code.
- Your output based on the data provided.

Sample Output:

```
Initial Books
   Book Id   Title                                 Customer Ck Out Dt  Late Fee Max Late
   ARTINT001 A.I. Practical Algorithms             333333   2015-11-15     0.30    55.00
   COBOLL001 How your Grandpa Coded in COBOL       NONE     0000-00-00     0.10    10.00
   EXCELL001 Excel at Excell                       444444   2015-09-01     0.80    65.00
   JAVADD001 Java Isn't an Addiction               777777   2016-02-01     0.30    60.00
   JOYPGM001 The Joys of Programming               NONE     0000-00-00     0.25    50.00
   JOYPGM002 The Joys of Programming               333333   2016-01-05     0.25    50.00
   JOYPGM003 The Joys of Programming               NONE     0000-00-00     0.25    50.00
   JOYPGM004 The Joys of Programming               NONE     0000-00-00     0.25    50.00
   LINUXX004 Learning Linux                        333333   2016-01-05     0.30    60.00
   PERLLL001 Is PERL the Jewel of Programming?     NONE     0000-00-00     0.10    10.00
   PRANKS001 Software Pranks                       NONE     0000-00-00     0.90    60.00
   PYTHON001 Learn Python Without Getting Bit      111111   2016-01-02     0.30    60.00
   PYTHON002 Learn Python Without Getting Bit      NONE     0000-00-00     0.30    60.00
   SQLDBB001 Making Your DB Queries SQueeL         555555   2016-02-16     0.30    60.00
   TECHDR001 My Technical Dream Job                NONE     0000-00-00     0.25    50.00

BOOK PPRINT
         TECHDR001
      SQLDBB001
         PYTHON002
   PYTHON001
         PRANKS001
      PERLLL001
         LINUXX004
JOYPGM004
         JOYPGM003
      JOYPGM002
         JOYPGM001
   JAVADD001
```

```
         EXCELL001
      COBOLL001
         ARTINT001
CUSTOMER BEGIN 111111 0.75 petem@xyz.net Pete Moss
CUSTOMER ADDRESS 123 Boggy Lane,New Orleans,LA,70112
******************** Library Receipt **********************
111111 petem@xyz.net Pete Moss (previously owed 0.75)
123 Boggy Lane
New Orleans, LA 70112
                                     Trans  Book       Date
CUSTOMER TRANS C JOYPGM001 2016-01-25
                                       C    JOYPGM001  2016-01-25
CUSTOMER TRANS C TECHDR001 2016-01-25
                                       C    TECHDR001  2016-01-25
CUSTOMER TRANS R PYTHON001 2016-01-25
                                       R    PYTHON001  2016-01-25   Late Fee =   2.70
CUSTOMER COMPLETE

                                                                    Total Fees =   3.45
CUSTOMER BEGIN 222222 0.00 pcorn@abc.net Pop Corn
CUSTOMER ADDRESS 456 Kernel,San Antonio,TX,78210
******************** Library Receipt **********************
222222 pcorn@abc.net Pop Corn (previously owed 0.00)
456 Kernel
San Antonio, TX 78210
                                     Trans  Book       Date
CUSTOMER TRANS C TECHDR001 2016-01-25
                                       C    TECHDR001  2016-01-25 *** Already checked out
CUSTOMER TRANS C TECHDR002 2016-01-25
                                       C    TECHDR002  2016-01-25 *** Book Not Found
CUSTOMER TRANS C JOYPGM004 2016-01-25
                                       C    JOYPGM004  2016-01-25
CUSTOMER COMPLETE

                                                                    Total Fees =   0.00
```