APS 105 — Computer Fundamentals
Lab #8: Linked Lists
Fall 2012

Important note: The course material necessary for completing this lab will be found in your lecture notes and in the Carter text up to the end of Section 10.4.

You must use the `submitaps105f` command to electronically submit your solution by 11:59pm on **Saturday, December 1, 2012**.

Objective

In this lab you will write a program that maintains information for an online bookstore, similar to the information on Amazon. The program will allow you to add and delete books from the bookstore inventory, to search the inventory for books by name or by rating, and to print out part or all of the book inventory.

The data in your book inventory will be stored in memory with the use of a linked list, with list nodes representing books in the inventory. Each node will contain members for storing a book's name (`char *`) and author (`char *`), its rating (`int`) and its category (`char`). There are four possible categories, each of which is identified by the first character in the words (F)iction, (H)istory, (P)hilosophy and (A)rts. Your linked list must be kept in a special order, with all the fiction books first, then the history books, then the philosophy books, and finally the arts books. If there is more than one book in the same category, then the books should be kept in order of their insertion (e.g., the last history book in the inventory should be the history book most recently inserted into the list; the first art book in the inventory should be the art book that was inserted first into the list, and so on). Books with duplicate names are not allowed in the inventory. You must detect such cases, disallow them and print an error. However, you can assume that the user enters a valid author, rating, and category.

Your program should be menu driven, with the user being offered a choice of the six commands described below:

Insert a new book into the bookstore inventory. The program should prompt the user for a book name and author name, a rating and a category. This information should be placed in a new node that has been created using the `malloc` function. And then the node should be inserted at the appropriate category in the linked list that stores the inventory data. Don't forget that the inventory must be stored in a special order, by considering the book's category first, and then (if needed) the order of insertion. If a node with the given name is already in the inventory, an error message should be produced and the new node should not be inserted into the linked list.

Delete a book from the bookstore inventory. The program should prompt the user for the name of the book to be deleted and then delete the node containing that name from the linked list that stores the bookstore inventory. If no book with the given name is found in the inventory, an error message should be produced.

Print the bookstore inventory, following a special order. Print the name, author, rating and category of each book, with each piece of information on a separate line. A blank line should be printed between each book. The special order assumes that the fiction books appear first, then the history books, then the philosophy books, and finally the art books. If a category has more than one book, then they should be printed in order of their insertion.

Search for a book using a name. The program should print the name of the book, as well as its author, rating and category, with each piece of information on a separate line. Search results should be printed using the same ordering defined for the *Print* function. If no book with the provided name is found in the inventory, an error message should be produced.

Search for books in the inventory that are rated higher than or equal to a given rating. The program should print the name, author, rating and category of any book that is rated higher than or equal to an input rating, with each piece of book information on a separate line. Search results should be printed using the same ordering defined for the *Print* function. A blank line should be printed between each book (if more than one is found). If no book in the inventory is rated higher than or equal to the given rating, an error message should be produced.

Quit the program. When the program is given the quit command, it should delete all of the nodes in the linked list by using calls to the `free` function. It should then try to print the empty linked list.

Your Task

Your task is to write a complete C program that maintains a bookstore inventory. Your C program must go in a file named `Lab8.c`.

To assist you in the production of your program, you will be given a file that contains a skeleton of a complete program. To copy the file to your home directory, run the command

```
cp /share/copy/aps105f/lab8/Lab8.c ~
```

The skeleton program includes all of the C statements required to implement the menu driven parts of the program. It also includes a few helpful functions for reading data and printing messages. All you need to do is implement the instructions for working with the linked list that stores the bookstore inventory.

We recommend that you take the following steps:

- Read the entire skeleton program carefully. Take note of the provided functions for reading strings, printing information about books, and for printing error messages. Using these functions will make it easier for you to satisfy the tester and marker programs.
- Figure out how to express the required book information in a node. Add a `struct` definition for a linked list node to the program.
- Add the function for inserting a node into the linked list. Your function will need to read the name, author, rating and category for the book. Test your program by trying to insert nodes into the linked list. Try to insert nodes with both new and duplicate names, and with different categories.
- Add a function for printing the linked list. Test your program by inserting entries into the linked list and then printing them out. Are the entries in the correct order?

- Add a function that searches the linked list for a book with the given **name**, and then either prints the appropriate entry, or if a book with the given name is not found, prints an error message.
- Add a function that searches the linked list for books that are rated higher than or equal to a given **rating**, and then either prints the appropriate entries, or if no node with a rating higher than or equal to the provided rating is found, prints an error message.
- Add the statements that need to be executed when the Quit command is entered. These statements should delete the linked list by using calls to the `free` function. To check your work, print the linked list after all of the elements have been deleted.
- Add a function for deleting a book. It will need to search the linked list for a given name, delete the appropriate node from the linked list and then use the `free` function to release the memory used to store the node. If the given name is not found in the bookstore inventory, print an error message.

We recommend that you test your program after attempting to complete each step. This way, if your program no longer works, you will know which statements are causing the error. Complete each step before moving on to the next one.

Marking

This lab will be marked out of 10. A marking program will be used to automatically mark your lab. The marking program will use the last version of the lab files you submitted using the `submitaps105f` command.

Full marks are given if your program works correctly, fewer if not, and zero if it cannot be compiled. Late submissions or submissions with an incorrect filename will result in a mark of 0 for the entire lab. The deadline will be strictly enforced, so avoid last minute submissions.

Any submitted solution that does not use linked lists will be assigned a mark of 0.

You can run a testing program, called `tester`, yourself to test the correctness of your solution. At the command line in your ECF account, run:

```
/share/copy/aps105f/lab8/tester
```

in the same directory as your solution file. The testing program will use a number of test cases to test your solution, and report success if your solution produces output that is identical to the expected output. Some of these test cases will be used by the marking program as well, but the marking program will also be using other test cases that are not included in the testing program to test the correctness of your program. This implies that even though you do not have access to the marking program, you will obtain at least partial marks if all of the test cases in the testing program report success with your solution.

Your mark for this lab will contribute 3% to your mark in the course.

What To Submit

When you have completed the lab use the command

```
    submitaps105f 8 Lab8.c
```

to submit your files. Make sure you name your file exactly as stated (including lower/upper case letters). Failure to do so will result in a mark of 0 being assigned. You may check the status of your submission using the command

```
submitaps105f -l 8
```

where `-l` is a hyphen followed by the letter *'ell'*. You can also download a copy of your submission by running the command

```
/share/copy/aps105f/lab8/viewsubmitted
```

Sample Output From Executing the Program

Here is a sample output from an execution of the program that you are to prepare. There is one space after each colon (`:`).

```
Online Bookstore Inventory Management Program.

Commands are I (insert), D (delete), S (search by name),
  V (search by rating), P (print), Q (quit).

Command?: I
  book: Nineteen Eighty-Four
  author: George Orwell
  rating: 5
  category: F

Command?: P

My Inventory:

Nineteen Eighty-Four
George Orwell
5
F

Command?: I
  book: Nineteen Eighty-Four
  author: Aldous Huxley
  rating: 4
  category: F

An entry for <Nineteen Eighty-Four> is already in the inventory!
New entry not entered.

Command?: I
  book: Brave New World
  author: Aldous Huxley
  rating: 4
  category: F

Command?: P

My Inventory:

Nineteen Eighty-Four
George Orwell
5
F

Brave New World
Aldous Huxley
```

```
4
F

Command?: I
  book: On the Shortness of Life
  author: Seneca
  rating: 5
  category: P

Command?: P

My Inventory:

Nineteen Eighty-Four
George Orwell
5
F

Brave New World
Aldous Huxley
4
F

On the Shortness of Life
Seneca
5
P

Command?: I
  book: The Trial
  author: Franz Kafka
  rating: 3
  category: F

Command?: P

My Inventory:

Nineteen Eighty-Four
George Orwell
5
F

Brave New World
Aldous Huxley
4
F

The Trial
Franz Kafka
```

```
3
F

On the Shortness of Life
Seneca
5
P

Command?: I
  book: A Short History of the World
  author: H. G. Wells
  rating: 3
  category: H

Command?: P

My Inventory:

Nineteen Eighty-Four
George Orwell
5
F

Brave New World
Aldous Huxley
4
F

The Trial
Franz Kafka
3
F

A Short History of the World
H. G. Wells
3
H

On the Shortness of Life
Seneca
5
P

Command?: Z

Invalid command.
Commands are I (insert), D (delete), S (search by
name), V (search by rating), P (print), Q (quit).

Command?: S
```

```
Enter the book name to search for: The Uncanny

The book with name <The Uncanny> is not in the bookstore inventory.

Command?: S

Enter the book name to search for: The Trial

The book with name <The Trial> was found in the bookstore inventory.

The Trial
Franz Kafka
3
F

Command?: V

Enter rating: 5

Nineteen Eighty-Four
George Orwell
5
F

On the Shortness of Life
Seneca
5
P

Command?: V

Enter rating: 6

No book(s) in the bookstore inventory is rated higher than or equal to <6>.

Command?: D

Enter the book name to delete: Brave New World

Deleting book with name <Brave New World> from the bookstore inventory.

Command?: P

My Inventory:

Nineteen Eighty-Four
George Orwell
5
F
```

```
The Trial
Franz Kafka
3
F

A Short History of the World
H. G. Wells
3
H

On the Shortness of Life
Seneca
5
P

Command?: Q

The bookstore inventory is empty.
```