COSC2430 Hw2: Library Book Management

(Linked List)

1. Introduction

You need to create a C++ program to manage a list of books. Each book will have attributes:

- book_id
- book_name
- author_name

You may need to add or remove books from the list. You may also need to sort the list of books.

2. Program and Input Specification

You will have an input file containing a list of books. Each line will be treated as a single book's information. Each book's information format will be:

- *book_id*:book id, *book_name*:book name, *book_author*:author name

In command.txt file, you will get corresponding command that you need to execute. You may get add, remove, or sort command. Command formats are,

- **add** *pos*:insert position, *book_id*: book id, *book_name*: book name, *book_author*: author name
- **remove** keyword:value
    - keyword is one of: *pos, book_id, book_name*, or *book_author*
- **sort** keyword
    - keyword is one of: *book_id, book_name*, or *book_author*

Your command file may contain multiple commands. Each line will have a single command. You will execute all commands sequentially one by one.

Assumptions:

- The input file is a small plain text file; no need to handle binary files.
- Each line will be terminated with a new line, '\n', character.

- Before executing any command, all records will be sequentially added from the input file.
- If you get same book name or author name, but not same book id, then you need to add the book.
- If you get an 'add' command, then you will add the book in the list at the provided position.
    - 0 is the starting index.
    - If you get index number that does not exist in the list, then you will add the record at the end of the list.
    - If you get existing book id, i.e., the book id is the same as an existing book id, you will not add the book.
- If you get a 'delete' command with book_name, or book_author keywords, then you will delete **all records** that matches with the book name or author name.
- If you get 'delete' command with pos keyword, then you will delete that position record.
    - Index starts from 0.
    - If you get invalid position, one that does not exist, then you do not need to delete any record.
- You may need to handle a **maximum 1000 records**.
- You may get a **maximum 100 commands** in a single command file.
- Book ID will be **always five-digit** number.
- If you get an empty line, then you will ignore it.
- Sorting will be **always in ascending order**. You need to sort the list based on the provided keyword.
- All records are case sensitive.

The main C++ program will become the executable to be tested by the TAs. The Result file should be written to another text file (output file), provided with the Command line.

Notice the input and output files are specified in the command line, not inside the C++ code. Notice also the quotes in the program call, to avoid Unix/Windows getting confused. See part 4.

3. Output Specification

The output file will contain the list of records after executing all commands. All records should be processed sequentially from beginning to end. Please, see examples for clarification about the format.

4. Program Execution:

The general call to the executable is as follows:
ListOperation "input=input.txt;command=command.txt;output=output.txt"

You can call the executable with another command line type,
ListOperation input=input.txt command=command.txt output=output.txt

5. Examples

Example 1 of input and output,

Input21.txt

book_id:32452, book_name:Data Structures Using C++, book_author:D. S. Malik
book_id:12452, book_name:Computer Networks, book_author:Peterson Davie
book_id:32432, book_name:Introduction to Algorithms, book_author:Thomas H. Cormen
book_id:32423, book_name:Algorithms in C++, book_author:Robert Sedgewick

Command21.txt

sort book_id
add pos:2 book_id:12456, book_name:Programming and Data Structures, book_author:xxx

Command line:

ListOperation input=input21.txt command=command21.txt output=output21.txt

output21.txt

book_id:12452, book_name:Computer Networks, book_author:Peterson Davie
book_id:32423, book_name:Algorithms in C++, book_author:Robert Sedgewick
book_id:12456, book_name:Programming and Data Structures, book_author:xxx
book_id:32432, book_name:Introduction to Algorithms, book_author:Thomas H. Cormen
book_id:32452, book_name:Data Structures Using C++, book_author:D. S. Malik

Explanation:

First your program will sort the book list, then it will add another book in the list at position 2, finally it will generate the output.

Example 2 of input and output,

Input22.txt

book_id:32452, book_name:Data Structures Using C++, book_author:D. S. Malik

book_id:12452, book_name:Computer Networks, book_author:Peterson Davie

Command22.txt

add pos:2 book_id:12452, book_name:Introduction to Algorithms, book_author:Thomas H. Cormen

remove book_author:Peterson Davie

Command line:

ListOperation input=input22.txt command=command22.txt output=output22.txt

output22.txt

book_id:32452, book_name:Data Structures Using C++, book_author:D. S. Malik

Explanation:

Book id '12452' matches with existing record, thus the book will not be added. The program will generate output after removing the book by matching the author name.

Example 3 of input and output,

Input23.txt

book_id:32452, book_name:Data Structures Using C++, book_author:D. S. Malik
book_id:12452, book_name:Computer Networks, book_author:Peterson Davie
book_id:12452, book_name:Introduction to Algorithms, book_author:Thomas H. Cormen
book_id:32423, book_name:Computer Networks, book_author:Robert Sedgewick
book_id:12352, book_name:Data Structures Using C++, book_author:D. S. Malik

Command23.txt

remove book_name:Computer Networks

Command line:

ListOperation input=input23.txt command=command23.txt output=output23.txt

output23.txt

book_id:32452, book_name:Data Structures Using C++, book_author:D. S. Malik
book_id:12352, book_name:Data Structures Using C++, book_author:D. S. Malik

Explanation:

Try to figure it out.

6. Requirements

- The output file should contain ONLY the output. Failure to adhere to the format will result in you not getting credit. Please ensure you follow the correct format.
- Timeout is set to 2 seconds.
- The provided test cases are designed to help you; however, they are not all inclusive. You should test your program with your own test cases to ensure your program is robust and can handle all cases. Keep in mind the 2 second timeout.

7. Hand over your homework

- Homework is an individual effort. Your homework will be automatically screened for code plagiarism against code from the other students and code from external sources. If you copy/download source code from the Internet or a book, it is better for you to acknowledge it in your comments, instead of the TAs detecting it. Code that is detected to be copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc.) will result in "Failure" for the course and being reported to UH administration.

- Homework 2 needs to be turned in to our Linux server, follow the link http://cs.uh.edu/~rizk/homework.html to get started with the server, argument manager, and other homework policies/info.

- Make sure to create a folder under your root directory, name it **hw2** (name needs to be lower case), only copy your code to this folder, no test case or other files needed. If you use ArgumentManager.h, don't forget to add it too.