

SciFiLi

Your final project is to create a library management program for SciFiLi (the Science Fiction Library) by applying all you've learned. Each of the following tasks is most efficiently accomplished by using a data structure, sometimes the same one and sometimes not. You need to decide which will be most appropriate for each task.

For each task, implement or use what you believe is the most appropriate data structure or algorithm AND JUSTIFY YOUR CHOICE IN THE COMMENTS. Part of your grade will be your justification.

1. The day starts with the program reading the file of books and storing them internally in a single data structure.
2. The librarian needs to be able to search by title or author.
 - a. If searching by author, output should list all books by that author.
 - b. The search should also tell if the book is checked out or not.
3. The librarian needs to be able to check in/out books.
 - a. Note: When people return books, they stick a pile in front of the librarian. Emulate the return process.
4. At the end of the day, when the librarian quits the program, output all books to a file (alphabetically by title) and if they're checked in or out.
5. If the library catches fire, the most important books need to be rescued first! Output a list of books (checked in only – no need to rescue a book that isn't there) that need to be rescued in order of importance.

Create a simple text-based or GUI menu to allow the user to select each task.

Requirements:

- You must use a minimum of 3 different data structures or algorithms learned this semester in the entire program (even if you did not write a program before using it).
 - The data structures must also be the versions you used this semester – so a binary tree needs to use nodes, not arrays. Ditto stacks and queues.
 - Yes, it can be something you've already written.
- You must also justify your choices.
- And your choices better be good ones. Don't pick a data structure or algorithm just because it's easy – pick it because it's the right tool for the job.
 - So a bubble sort on a linked list would be sad. Very sad.

About the input file:

Each book has one line in the text file showing all of the relevant information. For example:

Watership Down, AdamsR, 0, 7

As you can see, each piece of information is separated by “, “ and is in a specific order:

Title, Author, CheckInStatus, Priority

Title and Author are strings. CheckInStatus is a Boolean (1 = in, 0 = out). Priority is an integer, with 1 being the most important book in the library, 2 the next most important, etc.

And yeah, I know it's not all science fiction. I got the list from NPR's 100 sci-fi and fantasy everyone should read, then added a few more faves to it.

HOW THIS WILL BE GRADED:

- Syntax errors or other easily-corrected crashes will lose 10 points each occurrence. After the 3rd occurrence, you will receive an F.
- Each of the 3 data structures or algorithms required is worth 25 points:
 - Good choice for the problem at hand = 7 points
 - Correctly implemented = 10 points
 - Well-justified = 8 points
 - NOTE: If you use more than three, indicate in your comments which three you want me to consider for grading.
- There are 5 tasks to implement. Each is worth 5 points.