CSCI-1200 Data Structures — Fall 2019
Homework 7 — Library Maps

In this assignment you will design a data structure to maintain the holdings and checkout information for a library. By using STL's associative container object (`map`) we can make this system quite efficient and elegant. *Please carefully read the entire assignment before beginning your implementation.*

Your program will read from `std::cin` and write to `std::cout`, but we expect you will *redirect* the input (& output) to trick your program into reading & writing to files (see course webpage "Helpful C++ Programming Information"). Each line begins with a character signaling which of four operations should be performed:

`a 2 pride_and_prejudice` This operation will *add* the specified number of copies of the named book or dvd to the library. If the item is already in the library, the number of copies is incremented.

`c sally pride_and_prejudice` In this operation, a library patron is attempting to *checkout* an item from the library. If the item does not exist in the library, if no copies are currently available, or if the person already has a copy of this item checked out, the request is denied.

`r sally` With this operation, the library patron *returns* all of their checked out items to the library. A message indicating how many items were returned is output.

`l pride_and_prejudice` This operation *looks up* the specified item and outputs the number of copies available for checkout and the list of the patrons who currently have the item on loan. The list of patrons is in chronological order, with the oldest checkout first.

`p` This *prints* all patrons (in alphabetical order) and items they have on loan (in chronological order).

Examples of the messages your program must output are available on the course website. Please follow these examples exactly. To see if your program is performing perfectly, you should use the command line program, `diff`, which takes two files as arguments and outputs the differences between them. `diff` is available on GNU/Linux, WSL, and MaxOSX. Please see a TA or the instructor in office hours if you have a question about these programs.

You must carefully consider the performance of each of the library operations. Let n be the number of different titles in the library, m be the maximum number of copies of a given item, p be the number of patrons using the library, and c be the maximum number of items any patron has checked out at one time. All of the library operations should have sub-linear expected running time with respect to n. Furthermore, the `'c'` & `'r'` commands should have sub-linear expected running time with respect to p. *Hint: That means you should use maps. In fact, you'll need at least two of them!* In your `README.txt` file include the order notation for each operation in terms of n, m, p, and c. Create a diagram on your data structure design (following conventions from lecture for diagramming maps) and submit this diagram as a .png, .jpg, or .pdf.

You are not explicitly required to create any new classes when completing this assignment, but please do so if it will improve your program design. We expect you to use good coding style, including correct use `const` and pass by reference/alias as appropriate throughout your assignment. We have provided a partial implementation of the main program to get you started. You may use none, a little, or all of this, as you choose, but we strongly urge you to examine it carefully.

Be sure to make up interesting new test cases (both simple and complex) as you work. In your README.txt file describe your new test cases and your motivation for designing each. Submit these new test cases and the output from your program. (Omit from your submission any extra large test cases if it exceeds the submission file size limit, but be sure to describe them in your README.txt).

Please use the provided template `README.txt` file for any notes you want the grader to read. **You must do this assignment on your own, as described in the "Collaboration Policy & Academic Integrity" handout. If you did discuss the problem or error messages, etc. with anyone, please list their names in your `README.txt` file.**