**HW 7 Pseudocode**

* Each customer can only rent 1 costume at a time
* Input
  + a ghost 2
    - Adds specified number of copies of a particular costume type into the costume shop inventory
    - Name of the costume is a single string with no spaces (underscore used if the costume name is multiple words like Elvis\_Presley)
    - If the costume is already in the shop, the number of copies is incremented
    - Examples:
      * a ballerina 1 (Added 1 ballerina costume.)
      * a pirate 4 (Added 4 pirate costumes.)
  + r Sally Smith princess
    - Customer is attempting to rent a particular costume from the shop
    - Customer name is specified by two strings (first and last names in that order) and then the costume is specified by a single string
    - If the shop does not carry that type of costume OR if no copies are currently available OR if the person is currently renting a copy of this costume, decline request with appropriate message
    - If the person is currently renting a different costume, then the costume must be first returned, which will be done automatically by your code 🡪 messages indicating the returned costume (if any) and the rented costume are output
    - Example:
      * r Sharon Jackson pirate
        + Sharon Jackson rents a pirate costume.
  + l gorilla
    - This looks up the specified costume and outputs the number of copies available for checkout and the name of the customers who currently have this type of costume rented
    - Customers are printed in chronological order with the oldest rental first
    - If the shop does not carry this costume, then an appropriate message is printed
    - Example:
      * a ballerina 1
        + Added 1 ballerina costume.
      * a ballerina 2
        + Added 2 ballerina costumes.
      * l ballerina
        + Shop info for ballerina costumes:

3 copies available

* + p
    - This operation prints all customers who have ever attempted to rent from the shop (in alphabetical order by last name then first name) and the costume (if any) that they are currently renting
      * Examples:
        + a pirate 4
        + r Sharon Jackson pirate
        + p

Costume info for the 1 party attendee:

Sharon Jackson is wearing a pirate costume.

* README
  + Analyze the Big O time complexity using the following variables:
    - n = number of different costumes in the shop
    - m = maximum number of copies of a given costume
    - c = number of customers who visit the shop
    - All the operations should be sublinear running time with respect to n and the ‘a’, ‘r’, and ‘l’ commands should have sublinear running time with respect to c
  + Include in your README.txt file the order notation for each operation in terms of n, m, and c
* Data Structures Diagram (7 points of the HW grade)
  + Use this diagram to communicate design choices you made and how this data structure works
  + You can illustrate how a specific operation (just choose one) results in the insertion, modification, or removal of data in the structure
* Extra Credit
  + Reimplement the functionality of your program without maps (use lists or vectors instead)
  + Make your program take in an optional command line argument to specify the non-map version
  + How does the order notation for the expected performance of the program change?
  + Test the two versions of the program with larger datasets to confirm your predictions of the performance differences
  + Use the UNIX time command to measure the performance
  + Make up new larger test cases as necessary
  + Write up your analysis in your README.txt and submit both versions of the code (the map version should run by default with no command line arguments)

**You should use at least 2 maps in this assignment.**