CS180 Lab 9: Multi-Dimensional Arrays, Searching and Sorting
[Continued from lab 08]

Sections from the textbook relevant for this lab: 7.1, 7.2, 7.3, 7.4, 7.7, 8.2, 8.3
Lab for week: 10
Original lab created by: John Franklin Jr. Edited by: Dan Roberts

Learning Objectives

1. Using and manipulating multi-dimensional arrays
2. Simple search technique
3. Implementing a sort algorithm

Objectives

- Download and setup files and directories.
- Search through file for given last name and department.
- Sort file input by department.
- Sort file input by department and last name.

Files

Lab9.zip

Setup

Unzipping the lab file will make a new directory for Lab 9:

```
$ cd
$ cd cs180
$ unzip lab09.zip
$ cd lab09
$ drjava MedicalDatabase.java &
```

Download medical_database.txt and move the file to the lab09 that you just created.

Exercise 9.0: Preliminaries

Objective: Understand the differences between lab 8 and lab 9, and avoid implementing the same things twice. Become familiar with testing procedures, and with some Unix commands.

Steps:

1. Look over the lab assignment, the contents of the zip file, and the code we've provided for you. You might notice that it's very similar to last week's lab. However, there are a few key differences:
    a. The first portion of the lab (printing and formatting) has been omitted
    b. Some portions of last week's lab, such as reading the file from the disk and printing it, have been implemented for you.
    c. The test cases have been relaxed a bit
2. Take a careful look at the constructor for MedicalDatabase. You should notice that it handles reading the data from disk and storing it in the array. Because the constructor is the first method called for any object, you can assume that this will have happened when your methods are called, so you will not need to reimplement this functionality.
3. Look also at the way the main method uses the functions in MedicalDatabase. It's worth noting that when it wants to sort the array, it first calls the appropriate method and *then calls the printFormat() method*. In other words, your sort functions don't have to output anything, they just need to modify the `patients` array.
4. A note on the test cases: each one is just a bash script. If you open them up with your favorite text editor, you will see a list of Unix commands. If you were to type these commands one-by-one into the command line, you would get the same results – a script is just a convenient way to run the same thing more than once.
5. If a test case fails and you're wondering why, you can get a glimpse of your output and the desired output by running the command
   `$ head out out1`
   This will print out the first ten lines of the target output, followed by the first 10 lines of your output. You can use this to compare the two and look for differences. Type in `man head` for more information.

Exercise 9.1: Search for a patient

Objective: Search through file for a given last name and department.

Steps:

1. Find the `search(String lname, String department)` method and write your code inside this method.
2. Iterate through the `patients` array and look for an entry that matches the given last name and department. When you find a match, print the patient's information to the console, using the same format as in the `printFormat()` method (you can't quite copy and paste the line, but it's close).
3. To see a sample search, compile your code and run the following command:
   `$ java MedicalDatabase 3`
6. Run test case 3 to check your output against example code.

Exercise 9.2: Sort by department.

Objective: Use a simple sorting algorithm,, e.g., bubble sort, to sort each patient by department.

Steps:

1. Find the `sortByDepartment()` method and write your code inside this method.
2. Sort the `patients` array by department. You can implement any sort function you feel comfortable with, but we recommend a bubble or insertion sort.
3. To see the sorted results, compile your code and run the following command:
   `$ java MedicalDatabase 4`
4. Run test case 4 to check your output.

Exercise 9.3: Sort by department and name.

Objective: Use a simple sorting algorithm to sort each patient by department and by the last name.

Steps:

1. Find the `sortByDepartmentAndName()` method and write your code inside.
2. Sort `patients` by department and last name, so that the departments are in alphabetical order and the patients within each department are together and sorted by last name. The way you implement this is up to you.
3. To see the sorted results, compile your code and run the following command:
   `$ java MedicalDatabase 5`
4. Run test case 4 to check your output.

Test Cases

Type ./test<3-5> within the lab9 directory to check if you have passed test case <3-5>. For example type test1 within the lab9directory to check if test case 1 works.

Grading

Criteria	Percent
Test Case 3	50%
Test Case 4	40%
Test Case 5	10%

Turn In

If you are working from home, remember after you think your code is right, compile and then run on lore

To turn in first remove all the class files from your Lab09 folder:

```
% pwd
/u/u91/yourlogin/CS180/lab09
% rm *.class
```

Next change your current folder to your CS180 folder and run the turnin command.

```
% cd   ../
% turnin -v -c cs180=XXX -p lab09 lab09
```

Where XXX is your section number (ask your lab TA).

<End of lab 09>