Introduction to Programming - Lab Exercises for Week 17 (Trimester 2, week 4)

Some exercises with ArrayList<String>

1. This exercise is to redo Exercise 1 of last week using an ArrayList<String> instead of a String[].

   Here is some code to get you started:

   ```
   import java.util.ArrayList;
   public class LabEx1 {

      public static void main(String[] args) {
      String[] theUsers = {"Anne", "Tom", "Dick", "Kate", "Harry"};
      ArrayList<String> userList = new ArrayList<String>();
      for (String name : theUsers) {
            userList.add(name);
      }

      // Some of the code you are to write goes here


      } // end main()


         public static void find(ArrayList<String> list, String name) {

            // You also need to write some code here

         } // end find()

   } // end LabEx1 class
   ```

   Your program should read in a name from the keyboard and use the find() method above, which you should write, to output a message which says whether the name entered is in the list. See last week's handout to review the find() method that was supplied as an example, though if you did last week's lab your solution to exercise 1 will probably be of more help. Use a for loop to iterate over the list to search for the target.

2. Create a copy of the program you produced for question 1. Instead of using a for loop to search the list in the find() method, use the ArrayList<String> method, indexOf(). The indexOf() method takes a String as a parameter and returns the index position in the ArrayList<String> at which the name occurs, or -1 if the String is not present.

   Modify your main() method to call the new method to check whether the name is present and display a message that states whether the name is present.

3. Copy your solution to exercise 2 and extend it as indicated below.

   Modify your program so it maintains an ArrayList<String> of usernames. It should repeatedly show a menu with the following options:

   - Display the list of names
   - Read in a name and append it to the end of the list (but do not allow duplicate names, each name should be unique)
   - Read in a name and check whether it is in the list (as in exercise 2 above)
   - Read in a name and delete it from the list if it exists (hint: ArrayList has a remove() method).
   - Allow the user to change their username by reading in a name and changing it to a new name if it exists (hint: ArrayList has a set() method).
   - Exit the program

   Is this exercise easier or more difficult than the equivalent exercise 3 using arrays last week? Why?

More exercises using an array

4. In week 08 the first exercise asked you to find the number in the range 1 to 10000 that has the largest number of divisors. You only had to print out one such number. Find your solution to that lab exercise and revise the program so that it will print out all numbers that have the maximum number of divisors. If you cannot find your program, counting divisors is discussed in section 3.4.2 of the book and you can find the CountDivisors.java code from the book in the Week08 folder on Moodle. Use an array as follows: As you count the divisors for each number, store each count in an array. Then at the end of the program, you can go through the array and print out all the numbers that have the maximum count. The output from the program should look something like this:

   ```
   Among integers between 1 and 10000,
   The maximum number of divisors was 64
   Numbers with that many divisors include:
            7560
            9240
   ```

5. Suppose that temperature measurements were made on each day of 2014 in each of 10 cities. The measurements have been stored in an array

   ```
   int[][] temps = new int[10][365];
   ```

   where `temps[c][d]` holds the measurement for city number c on the dth day of the year.

To simulate this, write a program that generates random numbers less than 40 (or choose another range if you prefer, and you could make this different for each city) to represent the temperature in a given city on a given day and fill the array with these temperatures.

Once you have filled the array have the program print out the average temperature, over the course of the whole year, for each city. The average temperature for a city can be obtained by adding up all 365 measurements for that city and dividing the answer by 365.0.