Assignment 3 (100 points)

CS 113 Spring 2015

Due Date: See Syllabus

Name & UCID & Section Number: Please write your name, UCID, and section number on the submission – please staple together – no loose sheets

What is to be submitted?

1. Problem 1: Answers to the exercises
2. Problem 2: Program listing with the program output using the input in file `Data3.txt` and user interaction based on the input you have entered.

1. Exercises (25 points)

a) (20 points) Show the syntax (form) , explain its meaning, and give an example of each of

   i. `while` loop,

   ii. `for` loop,

   iii. `if` statement,

   iv. array (give an example declaration and its use), and

   v. boolean type

b) (5 points) What does a `break` statement do when executed inside a loop?

2. Directory[1] Lookup Application (75 points)

What is the assignment?

Write a directory (database) lookup application which operates by

1. reading a list of numbers stored in file `Data3.txt` to create a database of numbers:

[1] The word *directory* is used here as it use in "telephone" directory or a "store" directory" as opposed to its use in "file" directory which refers to the organization of files in a computer system such as Unix or Linux.

- The list of numbers is preceded by a number (first number in the list) that specifies the size of the list.
- This file name `Data3.txt` is supplied as a command-line argument to the application.

2. accepting user input (supplied at the keyboard) which are numbers that it checks to see whether or not they are in its database.

For each number `x` input by the user the program displays (prints) "`x is in the list`" or "`x is not in the list`" depending upon whether or not `x` is in the list.

The program terminates when the user enters Ctrl-Z (Ctrl-D) on the keyboard.

Input File `Data3.txt`

```
14
21
222
210
-44
78
88
96
0
-1
88
7890
567
999
-101
```

The list has 14 values as indicated by the first value in the file.

Sample user interaction (submit your version of the user interaction)

```
     Database Server is Ready for Number Lookups!
Number to Lookup: 33
     >> 33 is not in the database
```

```
Number to Lookup: -1
     >> -1 is in the database
Number to Lookup: -33
     >> -33 is not on the database
Number to Lookup: ^Z
Goodbye!
```

Hints for writing the program

1. Use the `ReverseList.java` program we studied in class as example for reading input from a file whose name is supplied as a command-line argument:
    a. check to make sure that a command-line argument is supplied
    b. set up a `Scanner` object to read from a file (file named `Data3.txt`)
    c. read the data in the file and store it in an array (named, for example, `list`)
2. Set up a `Scanner` object to read input from the keyboard (from `System.in`)
3. As long as there is input to be read, read input, see if the value read is in the array `list` or not and print an appropriate message.
4. You will need 3 loops
    a. one for reading data from the file,
    b. one for reading user inputs, and
    c. one for checking whether or not the value supplied by the user is in the list.

    The third loop will be nested inside the second loop. The first and third loops could be `for` loops like the one in `ReverseList.java` and the second one would be a `while` loop as in the example `copy.java` that we also did in class.
5. To see whether or not a value is in the list of numbers (stored as an array), you will have to use an `if` statement to compare values.
6. If you find that the value is in the list, then you need to exit from the nested `for` loop (third loop) using the `break` statement.
7. Upon exiting the third loop (either the loop completes by not finding the value we are searching for or it terminates because we found the value and we used a `break` statement to force an exit from the loop) we need to know whether or not the value was found. For this, before starting execution of the loop, we can use a `boolean` variable, say `found`, set to `false` indicating that we have not found the value as yet. Variable `found` is set to true in the loop if the value is found in the list (prior to executing the `break` statement).
8. The value of `found` controls the message printed that specifies whether or not we found the searched value.

Grading

The program will be graded as follow for a total of 75 points:

- Appropriate comments: 10 points
- Appropriately indentation: 10 points
- Reads the data from a file: 5 points
- Sets up an array of the correct size and stores the data in the array: 10 points
- Does the searches correctly: 30 points
- Good user interaction (asks user for input): 5 points
- Nicely formatted output: 5 points