CSCI 241 Data Structures
Programming Assignment 3
Electronic Turn-in due: 10pm, Wednesday Nov 30, 2016.

In this assignment, you will write a program that analyzes a data file (.txt file) containing baby names. This file is a comma-delimited file called yobYYYY.txt, which contains baby name records. Each record uses the format "name,gender,number," where name is 2 to 15 characters, gender is M (male) or F (female) and "number" is the number of occurrences of the name. The given file is sorted first on gender and then on number of occurrences in descending order. When there is a tie on the number of occurrences, names are listed in alphabetical order.

Your program must store the names and the counts of the names in i) a single binary search tree, ii) hashmap, and iii) array/linked list. Each name can only be stored once using the given data structure. **You must not use Java's build-in tree class and create your own tree class.** Your program must contain the following methods:

- **SearchName** for a name returns number of male and female babies born in that year who has that name. It should also return a rank for this name (how popular is this name for boys and girls).

- **MostPopularName** returns the most popular 10 male and female baby names for a given year with their numbers and percentage of babies with that name.

- **UniqueName** returns 5 male and female baby names that are unique with their frequency and percentage of babies with that name. For this method, ignore names that appear less than 5 times.

- **DisplayName** prints the names in alphabetical order, and next to each name prints the number of male and female babies that have this name and percentage of babies (male and female) for that name.

Development and Testing

The input .txt file will be available to you representing information about baby names and numbers.

- You will read the data file and create the data structure.

- When the program begins execution, your program should loop repeatedly and allow the user to ask questions preferred data structure and baby names. The user should enter two numbers – the first specifying which data structure the want (1 = tree, 2 = hash map, and 3= linear data structure) and the second specifying which information s/he wants to know (1 = Search for a name, 2 = Most-Popular Name, 3 = Unique Name, and 4 = Display Name). For option 1, the user should be asked to enter a name. If there is a match for a given name, the program should display the name and number of male and female babies sharing that name. If no such name is found, a descriptive message should be printed (i.e., this name does not exist in the name database) and the program would stop.

Examples: If the user enters 1 and 1 (option 1) and then as search name enters Rayan, then the output might look like the following:

```
Selected Data Structure: Tree

Selected Name: Rayan

Year     Male  Rank-Male Female Rank-Female

2014      439   606         53      3113
```

Output for 2 and 2 **MostPopularName** might look like the following:

```
Selected Data Structure: Hash Map

Year: 2014
```

Female Name	Frequency	%	Male Name	Frequency	%
Emma	20799	20%	Noah	19144	25%
Olivia	19674	19%	Liam	18342	24.5%
Sophia	18490		Mason	17092	
Isabella	16950		Jacob	16712	
Ava	15586		William	16687	
Mia	13442		Ethan	15619	
Emily	12562		Michael	15323	
Abigail	11985		Alexander	15293	
Madison	10247		James	14301	
Charlotte	10048		Daniel	13829	

Similarly, UniqueName should follow similar pattern for displaying unique names.

Points

This assignment will be scored by taking the points earned and subtracting any deductions. You can earn up to 55 points:

Component	**Points**
SearchName	5
MostPopularName	15
UniqueName	10
DisplayName	10
Write-up and Test Cases	5
Contribution Summary	5

Submitting Your Work

By 10 PM on the due date, you or your partner should submit **all files (everything that is required to compile and run your program)** and the following materials

1. Your write-up
2. Your test files (at least two different test files)
3. Contribution Summary (each student should email separately discussing your own and your partner's contribution)

While evaluating, we will compile all .java files, run it against a series of test cases, analyze your code, and read your write up.

Write-Up & Test Cases

In one or two pages, provide a write-up of your implementation. Please submit your write-up as a plaintext file named writeup.txt. Your write-up should include the following points:

1. Both your names
2. An acknowledgement and discussion of any parts of the program that are not working. Failure to disclose obvious problems will result in additional penalties.
3. An acknowledgment and discussion of any parts of the program that appear to be inefficient (in either time or space complexity).
4. A discussion of the portions of the assignment that were most challenging. What about those portions was challenging?
5. A discussion on how you approached testing that your program was correct and asymptotically efficient. What did test1.txt test? What did test2.txt test?

Academic Honesty

To remind you: you must not share code with anyone except your programming partner: you must not look at others' code or show your classmates your code. You cannot take, in part or in whole, any code from any outside source, including the internet, nor can you post your code to it. If you need help from other students, all involved should step away from the computer and discuss strategies and approaches, not code specifics. I am also available via email (do not wait until the last minute to email). **If you participate in academic dishonesty, you and your partner will fail the course**.