

CSCI 241 Data Structures
Programming Assignment 3
Electronic Turn-in due: 10pm, Wednesday Dec 2nd, 2015.

In this assignment, you will write a program that analyzes a set of five data files (.txt files) containing baby names. For each year of birth, the comma-delimited file called yobYYYY.txt contains baby name records. Each record uses the format "name,sex,number," where name is 2 to 15 characters, sex is M (male) or F (female) and "number" is the number of occurrences of the name. Each file is sorted first on sex 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 a single binary search tree for each year. You can use an additional binary search tree for storing data over the five years. Each name occurring in a year can only be stored once in the tree. 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.
- **MostPopularName** returns the most popular 5 male and female baby names for a given year with their numbers and over the five years.
- **LeastPopularName** returns the least popular 5 male and female baby names for a given year with their numbers and over the five years.
- **DistinctName** returns the number of distinct names, distinct male and female names for a year and over the five years.
- **DisplayName** prints the names in alphabetical order, and next to each name prints the number of male and female babies that have this name for a year and over the five years.

Development and Testing

Five .txt files will be available to you representing information about baby names and numbers, one for each year.

- When the program begins execution, you will read the data files and create binary search trees based on the data.
- Once you construct the BSTs, your program should loop repeatedly and allow the user to ask questions regarding baby names. The user should enter one number specifying which information s/he wants to know (1 = Search for a name, 2 = Most-Popular Name, 3 = Least Popular Name, 4 = Unique Name, 5 = 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 for the different years and over the five year period. 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 (option 1) and then as search name enters Rayan, then the output might look like the following:

Name: Rayan

Year	Male	Female	Total
2014	439	53	492
2013	357	45	402
2012	362	51	413
2011	308	40	348
2010	295	40	335
Overall	1761	229	1990

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

Year	F-1	F-2	F-3	F-4	F-5
2014	Emma, 20799	Olivia, 19674	Sophia, 18490	Isabella, 16950	Ava, 15586
2013	Sophia, 21147	Emma, 20876	Olivia, 18366	Isabella, 17573	Ava, 15211
2012	Sophia, 22267	Emma, 20902	Isabella, 19058	Olivia, 17277	Ava, 15512
2011	Sophia, 21816	Isabella, 19870	Emma, 18777	Olivia, 17294	Ava, 15480
2010	Isabella, 22883	Sophia, 20612	Emma, 17322	Olivia, 17012	Ava, 15418
Overall	Sophia, 104332; Emma, 98676; Isabella, 96334; Olivia, 89623; Ava, 77207				

Similarly, MostPopularName should also display top 5 popular male baby names. LeastPopularName should follow similar pattern for displaying least popular names for individual years and overall.

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	10
MostPopularName	10
LeastPopularName	10
DistinctName	5
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 email me **all your 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)

As a backup, you should also upload all the materials on Canvas.

While evaluating, I 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.