CIS1300 (Fall 2019) Assignment 3 (Part 1) – What's in a Name?

Introduction

There is an interesting data set created by the U.S. Social Security Agency that stores the most popular first names of babies born in the U.S. since the 1880's. You can information about it at https://www.ssa.gov/OACT/babynames/index.html. We will be working with the most popular names for male and female babies in each decade from 1880 to 2010. The information will be contained in 14 files with the following names:

1880Names.txt	1890Names.txt	1900Names.txt	1910Names.txt	1920Names.txt
1930Names.txt	1940Names.txt	1950Names.txt	1960Names.txt	1970Names.txt
1980Names.txt	1990Names.txt	2000Names.txt	2010Names.txt	

File Description

Each file contains 200 lines representing the top 200 names for male and female babies. The format of each line is as follows:

1 John 89,950 Mary 91,668

- **[1]** The first number is the rank, *i.e.* 1 means that these are the most popular male and female names in the decade from 1880 to 1889.
- **[John 89,950]** This is followed by the male baby name and then the number of babies with that name born in the decade.
- **[Mary 91,668]** The fourth item is the female baby name of rank 1, followed by the number of babies with that name.

Program 1: `babyQuery.c`

Source Code Files

Your program will have the name `babyQuery.c` and you will also use the header file `babies.h`. In `babies.h`, you will find the definitions that you will need for your program. It has the following contents:

```
/*  Defines */
#define MAXLENGTH 20
#define ROWS 200

/* Struct definitions */
struct pNames {
   int  year;
   int  rank[ROWS];
```

```
    char maleName[ROWS][MAXLENGTH];
    int  maleNumber[ROWS];
    char femaleName[ROWS][MAXLENGTH];
    int  femaleNumber[ROWS];
};

/* Function definitions */
int removeCommas ( char * );
```

You may add to this header file as needed but you cannot change want is already in the file.

Functionality

The program will accomplish the following tasks:

- Read in all the information about a decade that the user requests, *e.g.* if the user wants information about the 1880's then you must read in the file `1880Names.txt`.
    - As part of the input process you will have to eliminate the commas that appear in the numbers in the input files, e.g. the string 89,950 has to be changed to 89950 before being sent to `atoi()`. This must be done in a function called `removeCommas()` which will take one parameter, a pointer to a character array. The function will return the number of commas removed from the string.
- Store this information in a structure that will be given to you in the header file `babies.h`.
- You will then ask your user questions that will allow you to find the following types of information:
    - For a given **rank**, what is the (male, female, both) **name**, *e.g.* in the 1880's, the female name of rank 1 is Mary.
    - The **top** 10 names (male and female) for the given decade.
    - Given a name (female, male or both), find the rank for the given decade.

Question Script

The questioning of the user must follow the following script:

```
$ ./babyQuery
```
What decade do you want to look at? [1880 to 2010]: *1880*
Would you like to see a rank, search for a name, or see the top 10? [rank, search, top]: *rank*
Now there are three different paths for questioning:
Path 1: rank
What rank do you wish to see? [1-200]: *2*
Would you like to see the male (0), female (1), or both (2) name(s)? [0-2]: *2*
Rank 2: Male: William (84881) and Female: Anna (38159) *if response is 2*
Rank 2: Male: William (84881) *if response is 0*
Rank 2: Female: Anna (38159) *if response is 1*

Path 2: search
What name do you want to search for? [case sensitive]: *Emily*
Do you wish to search male (0), female (1), or both (2) name? [0-2]: *1*
In 1880, the female name Emily is ranked 91 with a count of 3368. *if response is 1*
In 1880, the male name Emily is not ranked. *if response is 0 and the name is not found*
In 1880, the female name Emily is ranked 91 with a count of 3368 and the male name Emily is not ranked. *if response is 2 – the female name will always go first even if it is not found*
Path 3: top

1	John	89950	Mary	91668
2	William	84881	Anna	38159
3	James	54056	Emma	25404
4	George	47651	Elizabeth	25006
5	Charles	46656	Margaret	21799
6	Frank	30967	Minnie	21724
7	Joseph	26292	Ida	18283
8	Henry	24139	Bertha	18263
9	Robert	24074	Clara	17717
10	Thomas	23750	Alice	17142

Notice that the columns are lined up. The number of spaces between each column is not less than 3 and not more than 8. The number of spaces is not the point, the point is that the columns are aligned and look pleasing to the eye.

After the answer has been presented to the user the following questions will be asked:

Do you want to ask another question about 1880? [Y or N]: *Y*
If the response is *Y* then return to the question "Would you like to see a rank, search for a name, or see the top 10? [rank, search, top]: ".
If the response is *N* then ask the following:
Would you like to select another year? [Y or N]: *Y*
If the response is *Y* then return to the question "What decade do you want to look at? [1880 to 2010]: ".
If the response is *N* then terminate the program with the message:
Thank you for using babyQuery.

Error Checking

Error checking is extremely important when users are giving information to the program. For all of the questions asked of the user, you must check that the input is exactly what was asked for. Let us examine the various responses requested from the user:

- What decade do you want to look at? [1880 to 2010]:
    - The response must be 1880, 1890, 1900, 1910, 1920, 1930, 1940, 1950, 1960, 1970, 1980, 1990, 2000, or 2010. No other numbers are acceptable.
- Would you like to see a rank, search for a name, or see the top 10? [rank, search, top]:

    - The user must type in rank or search or top – all lower case and all spelled correctly and in full.
- What rank do you wish to see? [1-200]:
    - Only numbers between 1 and 200 are acceptable.
- Would you like to see the male (0), female (1), or both (2) name(s)? [0-2]:
    - Only the numbers 0, 1, or 2 are acceptable.
- What name do you want to search for? [case sensitive]:
    - The requested string is to be treated as case sensitive (names in the files have the first letter in upper case and the rest in lower case). If they enter a name that does not follow this format, the string is to be accepted as input but the program is to do nothing to "fix" the case and thus the request will fail.
- Do you wish to search male (0), female (1), or both (2) name? [0-2]:
    - Only the numbers 0, 1, or 2 are acceptable.
- Do you want to ask another question about 1880? [Y or N]:
    - The user is to respond with a single letter, either Y or N but it is to be treated in a case insensitive manner; i.e. y and n are acceptable.
- Would you like to select another year? [Y or N]:
    - The user is to respond with a single letter, either Y or N but it is to be treated in a case insensitive manner; i.e. y and n are acceptable.

If the user makes an error, the program is to give an error message and then repeat the question. The following are the error messages that are to be given:

- What decade do you want to look at? [1880 to 2010]:
    - Acceptable decades are 1880, 1890, 1900, 1910, 1920, 1930, 1940, 1950, 1960, 1970, 1980, 1990, 2000, or 2010. No other numbers are acceptable.
- Would you like to see a rank, search for a name, or see the top 10? [rank, search, top]:
    - Please type in rank, search, or top exactly as requested.
- What rank do you wish to see? [1-200]:
    - Only numbers between 1 and 200 are acceptable.
- Would you like to see the male (0), female (1), or both (2) name(s)? [0-2]:
    - Only the numbers 0, 1, or 2 are acceptable.
- What name do you want to search for? [case sensitive]:
    - No error message is needed for this question.
- Do you wish to search male (0), female (1), or both (2) name? [0-2]:
    - Only the numbers 0, 1, or 2 are acceptable.
- Do you want to ask another question about 1880? [Y or N]:
    - Only the single characters Y or N are acceptable.
- Would you like to select another year? [Y or N]:
    - Only the single characters Y or N are acceptable.