asgn06: name games

Objectives

- Use concepts we have covered: arrays, strings, loops and functions to parse text
- Write an algorithm
- Use git and GitHub
- Refactor code
- Upload to your web host

Write a program that processes the included example file that contains people's names, outputs some statistics and generates output.

Setting up

- Create a new folder called `web182/asgn06-names`. This is where you will store your code and text file.
- Create a file named `index.php` and store it in `asgn06-names`.
- You may create other files -- function library or similar included files as needed. Be creative.
- Update your `web182/index.html` file.
- Download the data file in Moodle and store it in `asgn06-names`.

Git

Open your terminal and navigate to the following path. Replace my username with your own.
`bash cd /Users/charleskwallin/.bitnami/stackman/machines/xampp/volumes/root/htdocs/web182/asgn06`

Initialize git

```
git init
```

Work on the Master branch

Since you are just getting started with git, it is easiest to work on the master branch.

Add a README.md file

Files ending in `.md` or `.markdown` are called markdown files. Markdown is an HTML preprocessor language. We will cover more of it in class.

Create a file named `README.md` and past the following contnet.

```
#asgn06 names

This program is an exercise for school to practice PHP concepts: arrays, functions and stri
ng manipulation.
```

Save the file.

Make your first commit

The first time you add files to the staging area, you can use the dot as in `git add .`

```
git status
git add .
git commit -m "Original commit"
```

GitHub

Now that your code is at a good starting point, it is time to push it to your GitHub account

- Log into GitHub
- Click on the green, New button
- Name your repo `asgn06-names`

Make commits as needed

Once you are at a good breaking point then it is time to make a commit by following the instructions from above. Instead of using the dot operator to include all files, it is a better practice to list the files you have changed. For instance, let's say that I have only worked on the `index.php` file. I would do the following

```
git status
git add index.php
git commit -m "Updated index.php. Added coce for unique first names"
```

Since you are working locally with git, you do not need to continually push the code to GitHub; however, there is no harm in doing so if you would like the practice.

Input

For your test, use the data file in Moodle.

Output

1. The unique count of full names (i.e. duplicates are counted only once)
2. The unique count of last names
3. The unique count of first names

4. The ten most common last names (the names and number of occurrences)
5. The ten most common first names (the names and number of occurrences)
6. A list of 25 specially unique names (see below for details)
7. A list of 25 modified names (see below for details)

Name Validation

Here are some rules about what you should consider a name:

1. Names start at the beginning of the line.
2. Names follow these rules:
    - Formatted "Lastname, Firstname"
    - Contain only uppercase and lowercase letters

Names that don't follow those rules should be ignored.

Lists of Specially Unique & Modified Names

Take the first N names from the file where both of the following are true:

- No previous name has the same first name
- No previous name has the same last name

For example, consider these names:

```
Smith, Joan
Smith, John
Smith, Sam
Thomas, Joan
Upton, Joan
Upton, Tom
Vasquez, Cesar
```

These names would be part of the list of N names:

```
Smith, Joan
Vasquez, Cesar
```

These names would not:

```
Smith, John     # Already saw a last name "Smith"
Smith, Sam      # Already saw a last name "Smith"
Thomas, Joan    # Already saw a first name "Joan"
Upton, Joan     # Already saw a first name "Joan"
Upton, Tom      # Already saw a last name "Upton"
```

Your program must support an arbitrary value for N, but for your example output you may use 25.

Once you have this initial list of 25 names, print it. Then print a new list that contains 25 modified names. These

modified names should only use first names and last names from the initial 25 names. However, the modified list and the initial list should not share any full names, and no first or last name may be used more than once.

For example, if the initial list contains the names:

```
Brutananadilewski, Carl
Crews, Xander
Cartman, Eric
.. 22 more names ..
```

Then this is a valid output:

```
Brutananadilewski, Eric
Crews, Carl
Cartman, Xander
.. 22 more names ..
```

But this is not (because "Barney" and "Bambam" weren't in the initial list):

```
Brutananadilewski, Eric
Crews, Barney
Cartman, Bambam
... 22 more names ...
```

This is also incorrect (because "Cartman, Eric" is unchanged):

```
Brutananadilewski, Xander
Crews, Carl
Cartman, Eric
.. 22 more names ..
```

This is also incorrect (because "Carl" is used multiple times):

```
Brutananadilewski, Xander
Crews, Carl
Cartman, Carl
.. 22 more names ...
```

Other Notes

The program should provide the correct answer! First and foremost you should focus on finishing and getting the right answer.

The program should be as time and memory efficient as you can make it.

The program should be as concise and readable as possible.

Brief comments explaining implementation choices are welcome, but not required.

You can hardcode the name of the input file and the modified name count into your program (to make it easier to write)

but it should otherwise be built to handle an arbitrary file with the same format as the example file.

Being fancy and clever is fine, but not at the expense of finishing, getting the right answer, and having readable code.

What to Submit

Push your code to your webhost so the user can click on the `asgn06-names` link and see your results

Push your final code to GitHub so your instructor can read it

Add links to your website and GitHub account in the **Comments** section in Moodle.

It is not over...

We will revisit this code during the term to try and improve its effiiciency and organization. We may even try making a pull request from GitHub (time will tell!)