Fall 2020
Computer Science I — Maíra Marques Samary PhD
Section 5 – Tuesday and Thursdays – 12:00PM – Fulton 250

TA's Office Hours – online

James Monahan - monahajm@bc.edu - https://bccte.zoom.us/j/2822792652
Tuesdays 7:00 PM – 8:00 PM
Wednesdays 4:00 PM – 5:00 PM

Jennifer Joseph - - josephjz@bc.edu - https://bccte.zoom.us/j/5882755193
Wednesdays 11:00 AM – 12:00 PM
Thursdays 3:00 PM - 4:00 PM

Liam Murphy- - murpaue@bc.edu - https://bccte.zoom.us/j/3085424208
Tuesdays 2PM-4PM

Discussion Groups:
CSCI100701 - Tuesday 6:00 PM – 6:50 PM- Fulton Hall 220 (James Monahan)
CSCI100702 - Thursday 5:00 PM – 5:50 PM – Fulton Hall 220 (Jennifer Joseph)
CSCI100703 - Wednesday 4:00 PM – 4:50 PM – Gasson Hall 203 (Liam Murphy)

Assignment 5

Due date – 11/06/20 11:59 PM

General Instructions

Create a folder named ***HW5_LASTNAME_FIRSTNAME***. You will populate the folder with **ALL** of the .py files you write for this homework. To submit the homework, verify the folder includes all your .py files, compress (zip) the folder then upload to Canvas. Remember to include the following comments at the **top of each** of your .py files:

```
# author:
# assignment:
# description:
```

What to submit in Canvas?

Make sure all your files are saved in the folder HW5_LASTNAME_FIRSTNAME, then compress (zip) the folder and upload to Canvas.

If you encounter any problems in completing the assignment or in the submission process, please don't hesitate to ask for help. The sooner, the better!

Problem

Let's tackle a political question. It is standard political dogma that Democrats "want big government" and that Republicans "create jobs." Let's see if data supports the dogma. The official source of employment statistics is the U.S. Bureau of Labor Statistics (http://www.bls.gov/data/#employment) and we have collected data for private employment and government employment. Our assumption is that if a party is creating jobs, then private employment will increase, and if a party is creating bigger government, then government employment will increase. Your job is to extract that information from the files provided.

Specification

There are two employment data files provided. Both have comma-separated data. Look at the headings to understand their format. Numerical values are in thousands.

- *government_employment.txt*
- *private_employment.txt*

There is also a comma-separated file on presidents, their years, and their political party.

- *presidents.txt*

Since the transition from one term to another occurs partway through January, the last year listed for any president is the same as the first year of the next president. To keep things simple let's, count January entirely for the incoming president. That is, in the file the last year listed for a president will not count.

For example, George W. Bush's last year is listed as 2009 but he was president for only a few weeks that year so we will not count him as being president in 2009.

Also, watch out for the "Jr." for President Carter when you are reading the file—there is an extra comma.

What you have to do:

1. Your program will prompt for the file names. Use exceptions (or a message) to check that each file was opened without an error.

2. All president data can only come from the presidents.txt file, i.e. you cannot code specifics about presidents into your program. We will test your program using a file of that format, but with different entries, for example we may test on a file that has three of the lines of that file.

3. **As an academic requirement, your program must define and use at least two functions in a meaningful way. (Feel free to use more functions; I did.)**

4. Calculate and display in columns (see sample below):

    a) the average monthly private employment for each political party

    b) the average monthly government employment for each political party

    c) the private employment of the first month and last month of each president

    d) the change in private employment from the first month to the last month of each president;

Hint:

There is a useful formatting type for printing percentage

```
print("{:6.2%}".format(1/3))
```

33.33%

There is a useful formatting type that puts commas in numbers (either int or float):

```
print(“{:,}”.format(123456})
```

123,456

Sample Output

(note that we will test on the provided president’s file as well as a subset of the file)

```
Government employment average per month (millions)
Republican:     18,562
Democratic:     19,599


Private employment average per month (millions)
Republican:     92,225
Democratic:     96,700


Private Employment by president (millions)
President        First Month     Last Month      Difference      Percentage
Carter            64014           63072           -942            -1.47%
Reagan            63537           76219           +12682          19.96%
Bush Sr.          76663           86317           +9654           12.59%
Clinton           86393           98856           +12463          14.43%
Bush Jr.          98853           109040          +10187          10.31%
Obama             109208          108485          -723            -0.66%
Trump             108562          118690          +10128           9.33%


Government Employment by president (millions)
President        First Month     Last Month      Difference      Percentage
Carter            14090           14946           +856             6.08%
Reagan            14969           16008           +1039            6.94%
Bush Sr.          16010           17347           +1337            8.35%
Clinton           17365           19466           +2101           12.10%
Bush Jr.          19450           21546           +2096           10.78%
Obama             21538           22266           +728             3.38%
Trump             22264           21902           -362            -1.63%
```

report the change as both a difference and as a percentage

RUBRIC

	Excellent (100% of points))	Average (60% of points)	Needs Improving (40% of points)	Possible Points
Function module	• Functions are implemented properly • Function headers are complete and accurate	• Functions are generally implemented properly, but exhibit minor errors. • Function headers are generally complete and accurate, but some minor details are missing.	• Functions are implemented improperly. • Function headers are sketchy or missing.	**3**
User interface	• UI formatting is appropriate. • Prompts are complete and concise.	• UI formatting exhibits minor flaws.	• UI formatting is sketchy or haphazard.	**1**

	• Information is presented in a meaningful form.	• Prompts are not completely clear and concise. • Information presentation is slightly confusing.	• Prompts are confusing or missing completely. • Information presentation is completely confusing.	
FINAL SCORE				**10**