CPSC 1301, Computer Science I Quiz

Quiz 9a

This is a timed test. You have thirty minutes to complete the test. Your deliverable will be a plain text file, that is, an ASCII file with a `.txt` file extension. When you finish the test, upload your deliverable to Canvas. Do not publish your answer to your `git` repository.

You may work with your study partner for this quiz. In fact, working together is strongly encouraged. If you work with a partner, you must each make a seperate submission for credit, but you must also include the names of both authors in your submission.

1 Instructions

You will find a text file named `presidents_long.csv` in the Quizzes directory. This is a comma separated value (CSV) file. The contents of this file should be familiar to you from previous quizzes. Each row of the file contains a record of a U. S. President, shown by the header of the file:

```
"order","firstname","middename","lastname","birthstate","birthyear","birthmonth","birthday",
"officeyear","officemonth","officeday","deathyear","deathmonth","deathyear","religion","party"
```

You will write a short script that will (1) read the file contents into your script, and (2) munge the contents into a data structure described below. This mimics retrieving data from an external database and using the data to do useful work.

Your output will consist of one Python list. The list will contain anonymous dictionaries, one for each row in the data file. Each dictionary will consist of these items:

order This contains one integer, the order of the president, i.e., 1, 2, 3, . . .

name This contains a dictionary of three key/value pairs: `firstName, middleName, lastName`.

dates This contains a dictionary of three key/value pairs: `birth, office, death`. The value for each item contains a *list* of integers, consisting of the appropriate year, month, and day.

facts This contains a dictionary of three key/value pairs: `state, party, and religion`.

Note that you will have to split the data, using `string.split()`. You will have to remove the extra new line characters. You will have to remove the extra double quotes (”) using `string.replace()`. You will also have to convert strings to integers using `int()`.

Finally, you can use multiple methods for opening and reading files. I would like for you to use the File Context Manager, as shown on page 294 of your book. I have copied the code below. I have a very specific reason for this requirement, which I will explain to you if you ask. (Don’t worry, it’s a real reason.)

```
with open('data.csv') as myfile: # See Chapter 34 for details
    for line in myfile:
        ...use line here...
```

For those of you who have been reading the book, you will notice that we have been covering CSV and JSON formats. The XML format is coming up. These topics are advanced for this class, but at least you will gain experience with these formats even though you many not have been formally introduced to them.

2 Output

Your output should include the name of the quiz and a list as indicated above.

```
this is quiz 09a
{'order': 1, 'name': {'firstName': 'george', 'middleName': '', 'lastName': 'washington'}, 'dates': {'birth': [1722, 2, 22],
    'inaugurated': [1789, 4, 30], 'death': [1799, 12, 13]}, 'facts': {'state': 'virginia', 'party': '', 'religion': 'deist'}}
{'order': 2, 'name': {'firstName': 'john', 'middleName': '', 'lastName': 'adams'}, 'dates': {'birth': [1735, 10, 30],
    'inaugurated': [1797, 3, 4], 'death': [1826, 7, 4]}, 'facts': {'state': 'massachusetts', 'party': 'federalist', 'religion': 'unitarian'}}
{'order': 3, 'name': {'firstName': 'thomas', 'middleName': '', 'lastName': 'jefferson'}, 'dates': {'birth': [1743, 4, 13],
    'inaugurated': [1801, 3, 4], 'death': [1826, 7, 4]}, 'facts': {'state': 'virginia', 'party': 'democratic-republican', 'religion': 'deist'}}
{'order': 4, 'name': {'firstName': 'james', 'middleName': '', 'lastName': 'madison'}, 'dates': {'birth': [1751, 3, 16],
    'inaugurated': [1809, 3, 4], 'death': [1826, 6, 28]}, 'facts': {'state': 'virginia', 'party': 'democratic-republican',
    'religion': 'episcopalian'}}
{'order': 5, 'name': {'firstName': 'james', 'middleName': '', 'lastName': 'monroe'}, 'dates': {'birth': [1758, 4, 28],
    'inaugurated': [1817, 3, 4], 'death': [1831, 7, 4]}, 'facts': {'state': 'virginia', 'party': 'democratic-republican',
    'religion': 'episcopalian'}}
{'order': 6, 'name': {'firstName': 'john', 'middleName': 'quincy', 'lastName': 'adams'}, 'dates': {'birth': [1767, 11, 7],
    'inaugurated': [1825, 3, 4], 'death': [1848, 2, 23]}, 'facts': {'state': 'massachusetts', 'party': 'democratic-republican',
    'religion': 'unitarian'}}`
```

3 Template

You should not be needing a template at this point. Here is a minimal template to get you started. Note that `process(line)` calls `getdict(record)`, and that the last function does (most of) the work. You do not have to use this.

```python
#quiz09a.py

print("this is quiz 09a")

presidents = []

def process(line):
    pass

def getdict(record):
    pass

with open('presidents_long.csv') as f:
        for line in f:
            process(line)

for p in presidents:
    print(p)
```