Homework 11
Python Scripting

Assigned: Friday, March 25, 11:00AM

Due: Friday, April 1, 11:00AM (Hard Deadline)

Submission Instructions

Submit this assignment on Gradescope. You must submit every page of this PDF. We recommend using the free online tool PDFescape to edit and fill out this PDF. You may also print, handwrite, and scan this assignment.

There may multiple answers for each question. If you are unsure, state your assumptions and your reasoning for why you think your answer makes sense.

1 Getting Started

Before we begin, install Python 3 along with the package installation tool pip. Python 3 is the newest version of Python (succeeding Python 2). Similar to how Ubuntu lets you install packages for your operating system with `apt-get`, the `pip` program will let you easily install, update, and remove Python libraries (also called "packages").

Python comes with a lot of neat features that make very cumbersome tasks in other languages trivial. In the following questions, provide a snippet of Python code that solves the problem. These questions are meant to get you familiar with the language; additionally, we recommend referencing the Python documentation or trying a Python tutorial if you're new to the language.

For the following questions, the number of lines rendered is only a suggestion on the approximate length of the solution. You are not required to stay within the limit or go beyond the limit.

1. A common file format is CSV, or "comma-separated values". Here is an example of a CSV file:

```
1,2,3
50,600,7000
35,65,95
```

If we read this file into a string, it would be hard to access the number at position $(1, 2)$, for example. We would like to write `csv_data[1][2]` to access the element at $(1, 2)$. This problem takes substantial work in C++ but can be done in one or two lines of Python.

Given a CSV string, parse it into a Python list. In this example, we are only dealing with one row of a CSV file.

```python
csv_raw = "1,2,3"
# Your code here:



print(q1_csv_data) # should print: ['1', '2', '3']
print(q1_csv_data[1]) # should print: 2
```

2. Our previous solution fails in a lot of cases, such as commas within a CSV element. Most commonly done operations (like a full-fledged CSV parser) are already solved in python through either built-ins or through one of the many packages. Check out the csv package.
   Given a CSV file (q2.csv), parse it into a Python 2d-list using the csv package.

```
# Your code here:








print(q2_csv_data[1][1]) # should print: 600
print(q2_csv_data[2]) # should print: ['35', '65', '95']
```

3. One powerful feature of Python is the list comprehension, which allows you to express complicated array transformations in very little code.
   Using a list comprehension, create a new list from the given list, which contains only the odd integers.

```
arr = [1, 2, 3, 4, 5, 6, 7, 42, 42]
# Your code here:



print(odd_nums) # should print [1, 3, 5, 7]
```

4. Dictionaries are Python's map data structure, similar to `map` and `unordered_map` in C++. They can be used to associate a key with a value.
   Pick your favorite three courses and create a dictionary called `fav_courses` associating the course code (such as "EECS-203") with the course name (such as "Discrete Math"). Then write `print_courses`, which takes a dictionary as a parameter and prints out each course code, a colon, and then the course name (on separate lines).

```
def print_courses(courses):
   # Print courses here:




# Create 'fav_courses' here:






print_courses(fav_courses)
# You should have printed something like this (order doesn't matter):
# EECS-203: Discrete Math
# EECS-376: Foundations of computer Science
# EECS-477: Algorithms
# (Although most of you will not pick so many math-heavy courses)
```

5. Now we're going to play with file I/O, so that we can write scripts to alter files. **Create a file called `courses.txt` with these lines**:

```
EECS-280,Introduction to Programming and Data Structures
EECS-281,Data Structures and Algorithms
EECS-388,Computer Security
EECS-398,Essentials for Computer Scientists
EECS-445,Machine learning
```

Modify your script above to read data from `courses.txt` and combine it with your favorite courses listed above, and then print it out using `print_courses`. You do not need to rewrite your definition of `print_courses` or `fav_courses` here.

```
# Your code here:











# Should print out about eight entries. If your favorite course was listed in
# the previous question and was also in courses.txt, you do not have to print
# it out more than once.
print_courses(fav_courses)
```

2 Scripts

Before beginning, use pip to install the Python package `sh`.

1. `sh` is a full-fledged subprocess interface for Python; namely, it will let you run bash commands in Python easily. **Use sh to write a script given a text file, print out all the words in decreasing order of frequency, along with how many times each word appears in Python.**

2. **Use `sh` to write a script to go through all directories in the current directory, identify if they are git repositories, and if so, run `git pull` on each one.**

3 Exploration

You'll find that libraries are plentiful and powerful in Python. However, there are so many that it can cause issues with dependencies and incompatibilities. To exemplify, if you were to give a stranger your new scripts from this project they might not have sh installed, or might have different versions of other packages set-up (e.g., csv). To solve this problem we can use a *package manager*, which will allow us to manage our packages intelligently. We're going to let you pick your management tool! We recommend either using Conda or pip & virtualenv, but you may use any other setup.

There are lots of articles online that will help you pick; here are two discussions: "What is the difference between pip and conda?" and "So you want to install a Python package.".

Package ideas:

Matplotlib beautiful plotting.

BeautifulSoup4 web scrapping.

Selenium automate web browser interaction.

Jupyter interactive python notebooks.

Pandas database tool.

Flask web server.

Requests web requests.

Pillow image processing.

NumPy numerical computation.

PyQT GUIs.

sklearn machine learning.

iPython improved python command line tool.

Arrow powerful time and date package.

Progressbar progress bars, as the name suggests.

Logging intelligent logging system.

What package management system are you using? Create a snapshot of your environment after getting it to work on all the previous scripts in this homework, paste your snapshot here. Why is this useful?

Find a Python package that solves a problem you're interested in. What's the package called? Provide an example code snippet of a cool problem it solves and explain what it does. We've provided a few packages we think are interesting, as potential options.