Homework 4
Due April 12, 2016

Submissions are due by 11:59PM on the specified due date. Submissions may be made on the Blackboard course site under the Assignments tab. Late submissions will be accepted for a 10% penalty each day (up to 48 hours). Make sure your name and FSUID are in a comment at the top of the file.

1 faculty_directory.py (50 points)

In this part of the assignment, your job is to create a script that will navigate to the FSU CS department website and print out a directory containing the telephone number, office location, email, and webpage of every faculty member in the department. The root page for the faculty listings is http://www.cs.fsu.edu/department/faculty/.

From this page, you can find links to every individual faculty page. Each individual faculty page lists the links that you need. However, the only link that you may hardcode into your file is the one above. All other information from this point must be retrieved using crawling and scraping methods. Here is some guidance:
1. First, try to gather all of the links to the individual faculty pages.
2. Navigate to the first page and scrape the data to be output to the user.
3. Repeat step 2 until all faculty information has been output to the user.
An example of what your output should look like is shown below.

caitlin@pymachine$ python faculty_directory.py
Name: Margareta Ackerman
Office: 264 LOV
Telephone: (850) 645-4029
E-Mail: ackerman [at cs dot fsu dot edu]
************************************************
Name: Sudhir Aggarwal
Office: 263 Love Building
Telephone: (850) 644 0164
E-Mail: sudhir [at cs dot fsu dot edu]
************************************************
Name: Theodore Baker
Office: N/A
Telephone: N/A
E-Mail: baker [at cs dot fsu dot edu]
************************************************
...

Missing information should be indicated with an “N/A” entry. Be careful of edge cases or inconsistencies – double check your output against the web pages!
2 imgur_battle.py (50 points)

Your task is to write imgur_battle.py, a module that pits two or more imgur users against one another in a battle to see who has the most imaginary internet points. At the very least, your module should contain the function battle(user1, user2, ...), which accepts any number of imgur usernames. If you are not familiar with imgur, you should visit imgur.com to become familiar with the image-sharing website.

As long as at least two usernames are provided, the battle() function should compare the sum of points received from comments made by the users and print the winner’s username. The user with the highest sum is the winner. When summing the comment points, only comments made on posts on which all users have commented should be considered. You may assume any username entered is a valid username.

This task will involve scraping imgur.com. A couple of things to note:

- All imgur user pages have the following address: http://imgur.com/user/username. In last portion, “username” should be replaced with the user’s actual username.
- Posts on which the user has commented are shown on the user page under the heading “Gallery Comments”. These listings also include the points obtained for the comment.
- Note that a user may have commented multiple times on the same post and that battling users may have multiple posts in common.
- User page comment listings are not loaded statically in the browser; you must be able to continually request more comment information as necessary. All comments are loaded dynamically by requesting pages of the form http://imgur.com/user/username/index/newest/page/num/hit.json?scrolling, where num is an integer starting from 0 and incrementing by 1 as necessary. When there are no more comments to be loaded, a request will return a page that contains nothing.

You are encouraged to use the Firebug plugin for Firefox, or some other similar tool, to open up a console as you navigate through imgur’s website. This will give you insight into how imgur dynamically loads the comments as you scroll. Manually point your browser to the dynamically loaded pages to see what kind of information is returned.

You will likely want to use the following modules: requests and json. An example run is shown below:

```python
$ python
>>> import imgur_battle
>>> imgur_battle.battle("LastAtlas", "catslikecats", "Tighe")
Username: LastAtlas
Username: catslikecats
```
Username: Tighe
All users have commented on these posts: 3YJ0k
User LastAtlas has a total of 3 points.
User catslikecats has a total of 12 points.
User Tighe has a total of 2 points.
The winner is catslikecats!

You must print the users participating in the battle, the posts on which all users have commented, and the total number of points each user has received for all comments on those posts. Lastly, print out the winner or, in the case of a tie, print out the users who have the highest number of points.

Please note that this is a creative exercise on your part. You will likely use a number of libraries, tools, and ideas we have discussed in class, but there are some aspects which we may not have covered in detail. You are free to use whatever methods you want to solve the problem. You may have to do a little research on your own to come up with a solution. Be aware that because the performance of your module will depend on connecting to imgur multiple times, your program may take a little while to process information for users with a large number of comments.