# ASSIGNMENT 1

CS 432 Web Science

Mackenzie Kerchner

Mackenzie Kerchner
ODU CS 432 Web science Spring 2019 Wednesday, January 30, 2019

# Contents

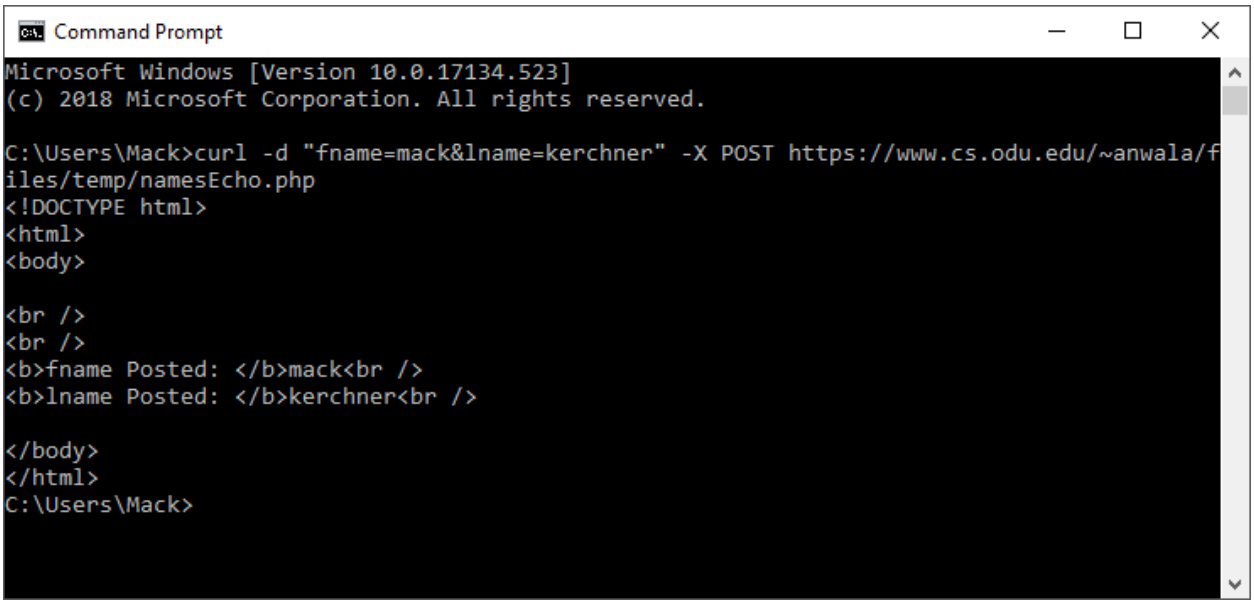
Mackenzie Kerchner
ODU CS 432 Web science Spring 2019 Wednesday, January 30, 2019

# Problem 1

```
1.  Demonstrate that you know how to use "curl" well enough to
correctly POST data to a form.  Show that the HTML response that
is returned is "correct".  That is, the server should take the
arguments you POSTed and build a response accordingly.  Save the
HTML response to a file and then view that file in a browser and
take a screen shot.

Feel free to use my simple server for sending POST
requests: http://www.cs.odu.edu/~anwala/files/temp/namesEcho.php
The server needs you to POST data for "fname" and "lname" fields.
```

## Solution

I googled curl command and read up on how it worked, https://curl.haxx.se/docs/manpage.html and https://gist.github.com/subfuzion/08c5d85437d5d4f00e58 were very helpful in figuring it out quickly.




curl -d "fname=mack&lname=kerchner" -X POST
https://www.cs.odu.edu/~anwala/files/temp/namesEcho.php
<!DOCTYPE html>
<html>
<body>

<br />
<br />
fname Posted: mack<br />
lname Posted: kerchner<br />

</body>
</html>

Mackenzie Kerchner


# Problem 2

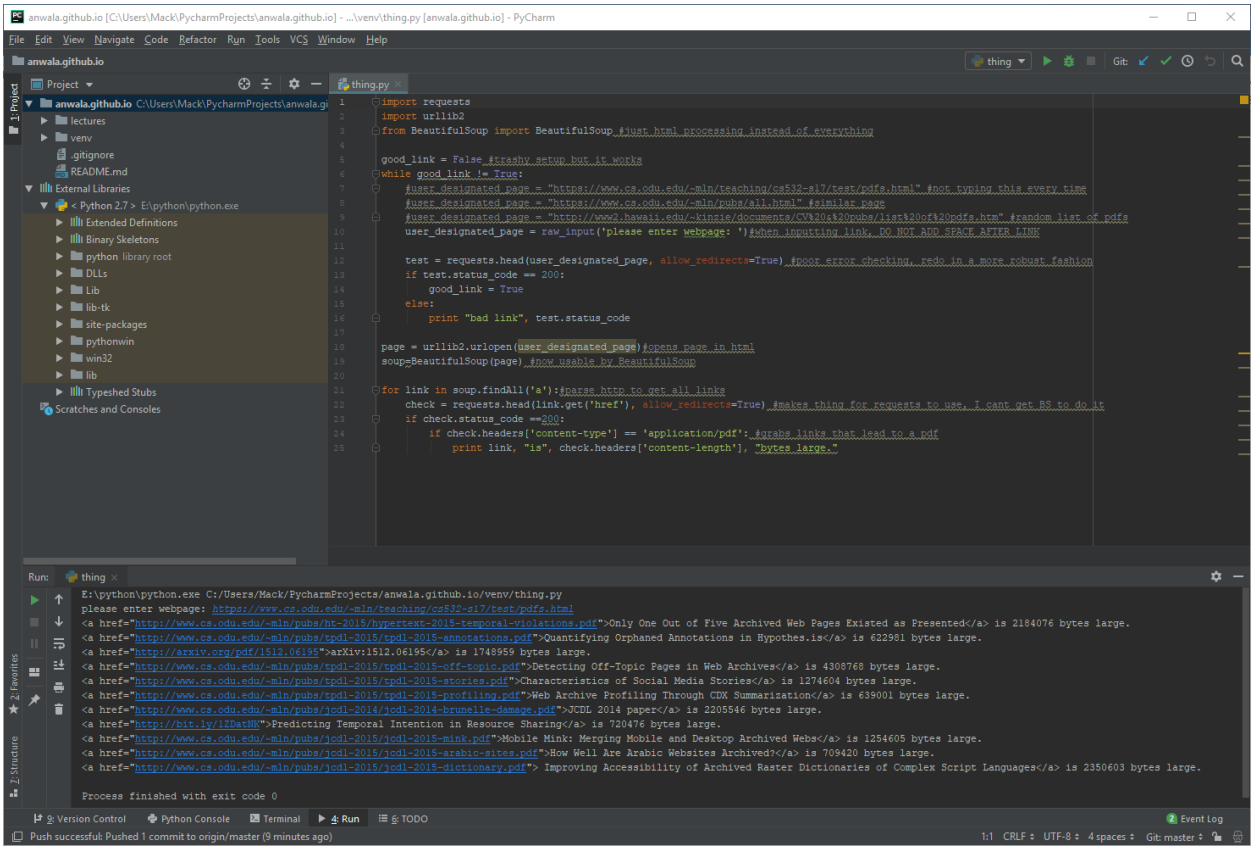
```
2.  Write a Python program that:
  1. takes as a command line argument a web page
  2. extracts all the links from the page
  3. lists all the links that result in PDF files, and prints out
     the bytes for each of the links.  (note: be sure to follow
     all the redirects until the link terminates with a "200 OK".)
  4. show that the program works on 3 different URIs, one of which
     needs to be:
     http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html
```

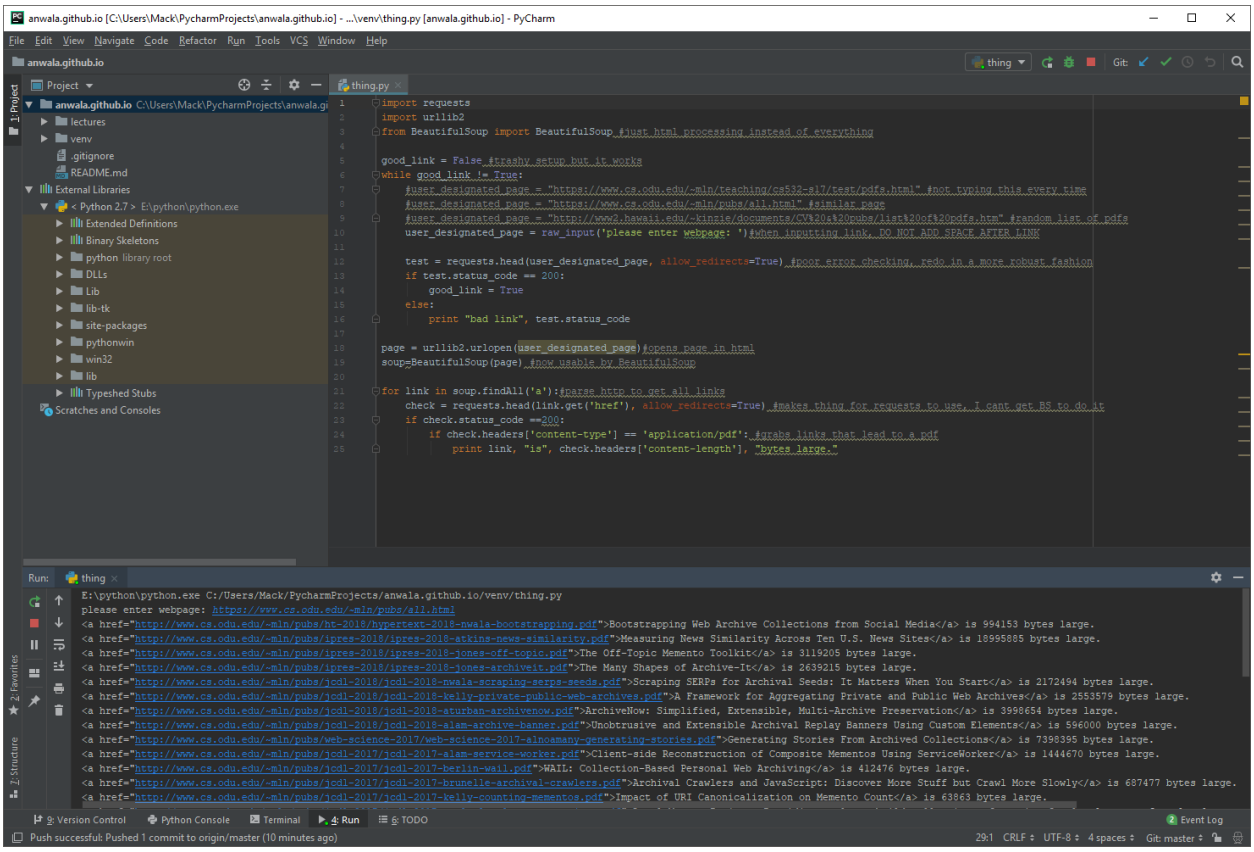
## Solution

initial pseudocode:
input webpage through command line
webpage opens y/n
    if n, return the error code
        loop until a page opens correctly
library to parse code until found all links
go through list of links
    {
    follow redirects until end
    pdf y/n
        if y, save link and number of bytes to list/tuple
    }
Print list of links and their associated bytes

This was my first time using python so the “python 101” [1] slideshow provided by Professor Nwala was invaluable in learning proper syntax. I used a combination of urllib3 [2] and BeautifulSoup [3] to handle data, the online user guides for the respective modules made it quite easy. I was getting an error with one of the links, which I found out in the lecture was being cause by redirects and solved this by setting allow_redirects=True, which is false by default, but only for HEAD. The program is not very robust and can break on some pages, I will have to put in more robust error checking so it works on a wider range of pages.

Mackenzie Kerchner
ODU CS 432 Web science Spring 2019 Wednesday, January 30, 2019



Mackenzie Kerchner

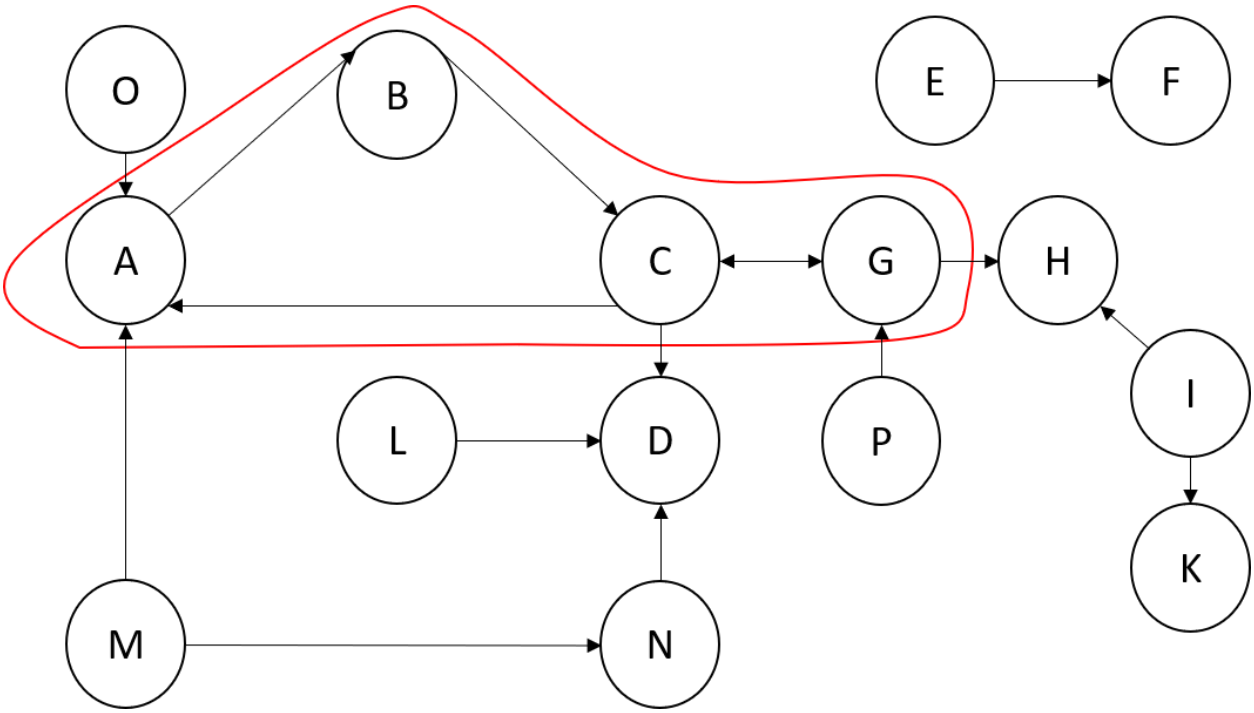

# **Problem** 3

```
Consider the "bow-tie" graph in the Broder et al. paper:
    http://snap.stanford.edu/class/cs224w-readings/broder00bowtie.pdf

    Many have found this link useful:
https://www.harding.edu/fmccown/classes/archive/comp475-s13/web-structure-
homework.pdf
    Now consider the following graph:

    A --> B
    B --> C
    C --> D
    C --> A
    C --> G
    E --> F
    G --> C
    G --> H
    I --> H
    I --> K
    L --> D
    M --> A
    M --> N
    N --> D
    O --> A
    P --> G

    For the above graph, give the values for:

    IN:
    SCC:
    OUT:
    Tendrils:
    Tubes:
    Disconnected:
```

## Solution

The solution to the problem is to create a graph of the connections, and then fill out the value sets with the criteria given on https://www.harding.edu/fmccown/classes/archive/comp475-s13/web-structure-homework.pdf. I did it SCC first, then IN and OUT, then tendrils, tubes, and disconnected pages.

Mackenzie Kerchner
ODU CS 432 Web science Spring 2019 Wednesday, January 30, 2019




SCC: A, B, C, G
In: O, M, P
Out: D, H, K
Tendrils: I, L
Tubes: N
Disconnected: E, F

A, B, C, and D are all strongly connected, as they can reach and be reached by any other page within the group. O, M, and P have no in-links and point to either SCC or a tube. D, H and K have no out-links and have in-links from either SCC or a tendril. I and L both have no in-links and point only to OUT. N has only in-links from IN and out-links to OUT. E and F are totally disconnected from the SCC.

Mackenzie Kerchner


**References**

[1] Python slideshow https://drive.google.com/file/d/16b7VpKcdaacDxcVlaLv_eV41xULVimH8/view. Accessed 1-30-2019

[2] urllib3 user guide https://urllib3.readthedocs.io/en/latest/user-guide.html Accessed 1-31-2019

[3] BeautifulSoup documentation https://www.crummy.com/software/BeautifulSoup/bs4/doc/ Accessed 1-31-2019