Assignment 1
CS532, Web Science, Spring 2017
Computer Science Dept
Old Dominion University

Hussam Hallak
CS Master's Student
Prof: Dr. Nelson
Due Date: 01/26/17

Question 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.

Answer:

To post data to a form using "curl" command, we need to use the "-X POST" option, and then add the option "-F" for each "field=value" in the form we want to post to.

Example:

Let's print HTML form a page using "curl". This page "index.html" is created for test purpose. It makes it possible to post data using a web browser:

```
root@ima-app:/var/www/Hussam# curl http://www.cs.odu.edu/~hhallak/532/A1/Q1/index.html
```

Output:

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

The page "welcome.php" expects data to be posted, name and email. Let's see what it looks like without data posted to it. The Superglobal array $_POST should be empty:

```
root@ima-app:/var/www/Hussam# curl www.cs.odu.edu/~hhallak/532/A1/Q1/welcome.php
```

Output:

```
<html>
<body>

 Array
(
)

</body>
</html>
```

Note: The code we use to print out the elements of the Superglobal array $_POST is what we see if we open "welcome.php" in a code editor:

```
<html>
<body>
<?php
print_r ($_POST);
?>
</body>
</html>
```

Now it's time to use "curl" command to post data to the page "welcome.php" and see how it works. The Superglobal array $_POST should contain our data that we posted:

```
root@ima-app:/var/www/Hussam# curl -X POST -F 'name=Hussam' -F 'email=me@hussam.us'
    http://www.cs.odu.edu/~hhallak/532/A1/Q1/welcome.php
```

Output:

```
<html>
<body>
 Array
(
    [name] => Hussam
    [email] => me@hussam.us
)
</body>
</html>
```

Now let's save our response to a file. We can easily do that by adding "-o" option, and add the name of the file where we want to save the response, to our previous "curl" command: **Note:** If the option "-O" is used, that is UPPERCASE O, there is no need to add a file name after it; the response will be saved to a file with the same page name.

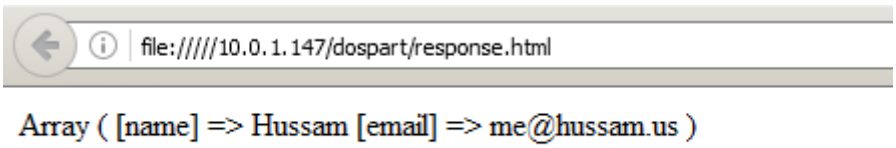
```
root@ima-app:/var/www/Hussam# curl -X POST -o response.html -F 'name=Hussam' -F
    'email=me@hussam.us' http://www.cs.odu.edu/~hhallak/532/A1/Q1/welcome.php
```

Output:

```
  % Total    % Received % Xferd Average Speed Time    Time     Time  Current
                                 Dload  Upload  Total   Spent    Left  Speed
100   348    0    97  100   251    999   2586 --:--:-- --:--:-- --:--:-- 3691
root@ima-app:/var/www/Hussam# ls
response.html
```

We can clearly see that the response is saved to a file named "response.html".

Screenshot: Let's open response.html is the browser to see the response:



Included Files:

index.html, welcome.php, response.html, screen_shot.png, session.txt

Question 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

Answer:

```
import sys
from bs4 import *
import urllib2
import re

if len(sys.argv) != 2:
   print "Usage: Python extracrPDF.py <url>"
   print "e.g: Python extracrPDF.py http://example.com/page.html"
else:
   url = sys.argv[1]
   print "Entered URL:"
   print url
   html_page = urllib2.urlopen(url)
   print "Final URL:"
   print html_page.geturl()
   print "********************"
   soup = BeautifulSoup(html_page, "html.parser")
   links = []
   for link in soup.findAll('a', attrs={'href': re.compile("^http://")}):
      links.append(link.get('href'))
   for link in links:
      try:
         r = urllib2.urlopen(link)
         if r.headers['content-type'] == "application/pdf" and r.getcode() == 200:
            print "Extracted link:"
            print link
            print "Extracted link final URL:"
            print r.geturl()
            print "Size: " + r.headers['Content-Length']
            print "-------------------------------------"
      except urllib2.HTTPError as e:
         print "There is an error extracting PDF files in this link:"
         print "Error Code:"
         print e.code
```

Running the program:

The program takes a link as a command line argument. It follows all the redirects until it terminates with a "200 OK". Then it begins to extract all links to PDF files and prints the PDF link, the final destination for the link, and file size.

First Test Case:

Let's test the link:
http://hussam.us
The link above is redirected to the link:
http://www.cs.odu.edu/~hhallak/
This page: http://www.cs.odu.edu/~hhallak/ does not contain any links to PDF documents.

```
root@ima-app:/var/www/Hussam# python extractPDF.py http://hussam.us
```

Output:

```
Entered URL:
http://hussam.us
Final URL:
http://www.cs.odu.edu/~hhallak/
********************
```

Required Test Case:

http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html

```
root@ima-app:/var/www/Hussam# python extractPDF.py
    http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html
```

Output:

```
Entered URL:
http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html
Final URL:
http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html
********************
Extracted link:
http://www.cs.odu.edu/~mln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
Extracted link final URL:
http://www.cs.odu.edu/~mln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
Size: 2184076
-------------------------------------
Extracted link:
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
Extracted link final URL:
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
Size: 622981
-------------------------------------
Extracted link:
http://arxiv.org/pdf/1512.06195
Extracted link final URL:
https://arxiv.org/pdf/1512.06195.pdf
Size: 1748961
-------------------------------------
Extracted link:
```

```
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf
Extracted link final URL:
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf
Size: 4308768
--------------------------------------
Extracted link:
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-stories.pdf
Extracted link final URL:
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-stories.pdf
Size: 1274604
--------------------------------------
Extracted link:
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
Extracted link final URL:
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
Size: 639001
--------------------------------------
Extracted link:
http://www.cs.odu.edu/~mln/pubs/jcdl-2014/jcdl-2014-brunelle-damage.pdf
Extracted link final URL:
http://www.cs.odu.edu/~mln/pubs/jcdl-2014/jcdl-2014-brunelle-damage.pdf
Size: 2205546
--------------------------------------
Extracted link:
http://bit.ly/1ZDatNK
Extracted link final URL:
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-temporal-intention.pdf
Size: 720476
--------------------------------------
Extracted link:
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
Extracted link final URL:
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
Size: 1254605
--------------------------------------
Extracted link:
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
Extracted link final URL:
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
Size: 709420
--------------------------------------
Extracted link:
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-dictionary.pdf
Extracted link final URL:
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-dictionary.pdf
Size: 2350603
--------------------------------------
```

Additional Test Case:

http://www.cs.odu.edu/~hhallak/pdfs/index.html

```
root@ima-app:/var/www/Hussam# python extractPDF.py
    http://www.cs.odu.edu/~hhallak/pdfs/index.html
```

Output:

```
Entered URL:
http://www.cs.odu.edu/~hhallak/pdfs/index.html
Final URL:
http://www.cs.odu.edu/~hhallak/pdfs/index.html
********************
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/A4.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/A4.pdf
Size: 191976
-------------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/cs772A2.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/cs772A2.pdf
Size: 2036216
-------------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/cs772all.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/cs772all.pdf
Size: 1297575
-------------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/des-s-boxes.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/des-s-boxes.pdf
Size: 90917
-------------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/hashes_message_digests.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/hashes_message_digests.pdf
Size: 126965
-------------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/introduction_authentication.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/introduction_authentication.pdf
Size: 50308
-------------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/introduction_cryptography.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/introduction_cryptography.pdf
Size: 55243
-------------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/introduction_general.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/introduction_general.pdf
Size: 26124
-------------------------------------
```

```
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/introduction_openssl.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/introduction_openssl.pdf
Size: 33765
-----------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/kerberos.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/kerberos.pdf
Size: 30051
-----------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/lectures.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/lectures.pdf
Size: 31128
-----------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/Number Theory.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/Number Theory.pdf
Size: 145137
-----------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/openssl.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/openssl.pdf
Size: 15481
-----------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/pem_smime.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/pem_smime.pdf
Size: 36764
-----------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/PKI_Certificates.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/PKI_Certificates.pdf
Size: 47536
-----------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/Primes.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/Primes.pdf
Size: 149857
-----------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/secret_key_cryptography.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/secret_key_cryptography.pdf
Size: 692743
-----------------------------------
Extracted link:
```

```
http://www.cs.odu.edu/~hhallak/pdfs/security_handshake.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/security_handshake.pdf
Size: 58099
-------------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/ssl_https.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/ssl_https.pdf
Size: 49048
-------------------------------------
Extracted link:
http://www.cs.odu.edu/~hhallak/pdfs/ssl_programming.pdf
Extracted link final URL:
http://www.cs.odu.edu/~hhallak/pdfs/ssl_programming.pdf
Size: 52201
-------------------------------------
```

Included Files:

extractPDF.py, README

Note:

The file README contains the following:

- How to use the program
- Required Python version
- Required Libraries

Question 3:

Consider the "bow-tie" graph in the Broder et al. paper (fig 9): http://www9.org/w9cdrom/160/160.html

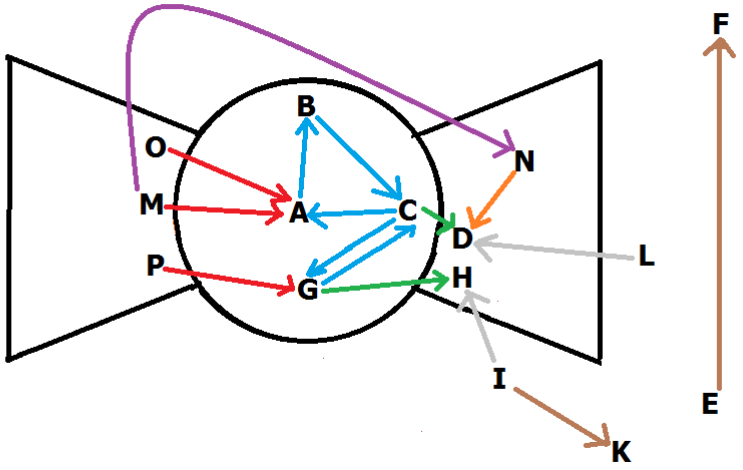
Now consider the following graph:

A $\longrightarrow$ B
B $\longrightarrow$ C
C $\longrightarrow$ D
C $\longrightarrow$ A
C $\longrightarrow$ G
E $\longrightarrow$ F
G $\longrightarrow$ C
G $\longrightarrow$ H
I $\longrightarrow$ H
I $\longrightarrow$ K
L $\longrightarrow$ D
M $\longrightarrow$ A
M $\longrightarrow$ N
N $\longrightarrow$ D
O $\longrightarrow$ A
P $\longrightarrow$ G

For the above graph, give the values for: IN, SCC, OUT, Tendrils, Tubes, Disconnected.

Answer:

IN: O, M, P
SCC: A, B, C G
OUT: H, D, N
Tendrils: I, K, L
Tubes: there is one tube from M to N (M $\longrightarrow$ N).
Disconnected: E, F



Included Files:

bowtie.png

References

[1] Python For Beginners. Available from World Wide Web:(http://www.pythonforbeginners.com/).

[2] Cambridge University Press. Available from World Wide Web: (http://nlp.stanford.edu/IR-book/html/htmledition/the-web-graph-1.html).