All files mentioned in this document should be uploaded into the *github* repository.

Problem 1

A shell script, called *ranking.sh*, is created by a Python program – *ranking.py*. This script contains commands that will achieve the following tasks:

- Downloading the 1000 URIs from the previous assignment using curl. See examples below:

```
$curl www.cnn.com > file1.html
$curl www.yahoo.com > file2.html
$curl www.alarabiya.com > file3.html
...
```

  The output files will have their names with sequential numbers (e.g. *file1.html*, *file2.html*, ...,*file1000.html*). A number in a file name indicates a specific URL. In other words, *file1.html* contains the raw *html* format of the first URI listed in the file *links.txt* while *file2.html* is the raw *html* format corresponding to the second URI in *links.txt* and so on.

- Remove (most) of the HTML markup from *.html* files and store results in files called: *file1.html.processed*, *file2.html.processed*, ..., *file1000.html.processed*. This can be done by the *lynx* command:

```
$lynx -dump -force_html file1.html > file1.html.processed
$lynx -dump -force_html file2.html > file2.html.processed
$lynx -dump -force_html file3.html > file3.html.processed
...
```

Problem 2

To count number of words, another two commands are added to the shell script *ranking.sh*:

- By *wc* command, a list of numbers of all words of all *.processed* files can be obtained. This list will be stored in a file called *wordsFreq.txt*. See the following examples:

```
$wc -w < file1.html.processed >> wordsFreq.txt
$wc -w < file2.html.processed >> wordsFreq.txt
$wc -w < file3.html.processed >> wordsFreq.txt
...
```

  The output of *wordsFreq.txt*:

```
1127
2259
1322
...
```

- By both *grep* and *wc* commands, we can get a list of numbers of occurrences of a term in all *.processes* files. In this assignment, I have chosen the term **song**. The output will be stored in a file called *termFreq.txt*:

```
$grep -rohiw Shakira file1.html.processed | wc -w >>
   termFreq.txt
$grep -rohiw Shakira file2.html.processed | wc -w >>
   termFreq.txt
$grep -rohiw Shakira file3.html.processed | wc -w >>
   termFreq.txt
...
```

  The output of *termFreq.txt*:

```
0
0
0
.
.
.
7
3
...
```

Because we are to choose only 10 documents containing the term, I have done all calculations shown in the below table manually.

- Number of Documents = 1000
- Number of Documents with the term song = 133

See table 1 for results.

Problem 3

Table 2 shows the estimation of the page rank of the 10 URI included in table 1 using the following page rank estimator:`http://www.seocentro.com/tools/search-engines/pagerank.html`. Because this tool always gives a page rank between 1 and 10, the result is divided by 10 to normalize the value to be from 0 to 1. Before even starting

Words in Doc.	Term Freq.	TF	IDF(song)	TFIDF	URI
836	8	0.010	2.911	0.028	http://www.youtube.com/watch?v=OFGgbT_VasI&feature=youtu.be
809	6	0.007	2.911	0.022	http://www.youtube.com/watch?v=2XMN2dg7OuU
873	6	0.007	2.911	0.020	http://www.youtube.com/watch?v=qjOeKRb6fco&feature=youtu.be
1194	4	0.003	2.911	0.010	https://www.youtube.com/watch?v=ln_RwnQC_vQ&feature=youtube_gdata_player
650	2	0.003	2.911	0.009	http://musiclikeneverbefore.com/index.html
1059	3	0.003	2.911	0.008	http://www.youtube.com/watch?v=c7Rd5rchoiI
1034	2	0.002	2.911	0.006	http://www.youtube.com/watch?v=xI44Xr2D0Ck&feature=youtu.be
1241	2	0.002	2.911	0.005	http://www.mjtunes.com/
1750	3	0.002	2.911	0.005	http://www.youtube.com/watch?v=s8QYxmpuyxg&list=PLEUun43OsA1egklY4LVnkO_Satgfy-TYO
3825	4	0.001	2.911	0.003	http://www.5pinkave.com/

Table 1: 10 Hits for the term : **song**

this experience, I was almost sure that the result produced from both mechanisms will be totally different since they use different algorithms to produce the page rank. I think the page rank estimators, mentioned in question 3, involve more complicated computations. I was surprised when seeing 50 percent of the results in table 1 and 2 are identical. URIs, placed in rows 1, 3, 4, 6 and 10 are the same pages in both tables! On the other hand, I can not recogize any pateren for the rest 5 pages. For eaxmple, The URL, placed in row 2 in table 1, is in row 8 in table 2 while the fifth URL in table 1 is placed row 9 in table 2.

Problem 4

As mentioned in 'http://stackoverflow.com/questions/2557863/measures-of-association-in-r-kendalls-tau-b-and-tau-c'

$KendallTau_b = (P - Q)/((n0 - n1)(n0 - n2))^{1/2}$

$Where$:

P: concordant pairs

Page Rank	URI
0.6	http://www.youtube.com/watch?v=OFGgbT_VasI&feature=youtu.be
0.6	http://www.youtube.com/watch?v=s8QYxmpuyxg&list=PLEUun43OsA1egklY4LVnkO_Satgfy-TYO
0.5	http://www.youtube.com/watch?v=qjOeKRb6fco&feature=youtu.be
0.4	https://www.youtube.com/watch?v=ln_RwnQC_vQ&feature=youtube_gdata_player
0.4	http://www.youtube.com/watch?v=xI44Xr2DOCk&feature=youtu.be
0.3	http://www.youtube.com/watch?v=c7Rd5rchoiI
0.3	http://www.mjtunes.com/
0.2	http://www.youtube.com/watch?v=2XMN2dg7OuU
0.2	http://musiclikeneverbefore.com/index.html
0.0	http://www.5pinkave.com/

Table 2: The page rank estimation for the 10 URIs included in table 1

Q: discordant pairs

N0: n(n-1) / 2

n1: the number of tied pairs on x

n2: the number of pairs pairs tied on y

TFIDF	Page Rank	Y pairs in natural order	Y pairs in reverse natural order
0.003	0.0	9	0
0.005	0.3	5	2
0.005	0.6	1	6
0.006	0.4	3	3
0.008	0.3	3	2
0.009	0.2	4	0
0.010	0.4	2	1
0.020	0.5	1	1
0.022	0.2	1	0
0.028	0.6	0	0

Table 3: to compute P, Q, X0 and Y0 values

From table 3 when can get the following:

P = 29

Q = 15

n0 = (10 * 9)/2 = 45

n1 = 2(1) / 2 = 1

N2 = (2(1) + 2(1) + 2(1)+ 2(1)) / 2 =4

$KendallTau_b = (29 - 15)/((45 - 1)(45 - 4))^{1/2}$

$KendallTau_b = 0.32$

$P = ((n*\sum_{i=1}^{n} xy) - (sum_{i=1}^{n} x * sum_{i=1}^{n} y))/(\sqrt{n(sum_{i=1}^{n} x^2) - (sum_{i=1}^{n} x)^2} * \sqrt{n(sum_{i=1}^{n} y^2) - (sum_{i=1}^{n} y)^2})$

P = 0.011