*Q1. (10 points) Write regular expressions that match the following patterns – This doesn’t have to be a program, submit this as a theoretical answer (create a pdf file with the answers):*

*a) An odd digit followed by an even digit*

*(eg. 14 or 38)*

***/[13579][02468]/***

*b) A letter followed by a non-letter followed by a number*

*(eg. A#1)*

***/[A-Za-z][^a-zA-Z0-9][0-9]/***

*c) A word that starts with an upper case letter and ends with some punctuation mark (`!’, `?’, `,’, `;’, `:’, etc.)*

*(eg. Microphone!)*

***/^[A-Z].\*[.!?,;:\\-]$/***

*d) The string "ping" in any combination of upper and lower cases letters*

*(eg. In a sentence such as “Mapping the address was complex,*

*so I pinged and pinged him time and again, griping. Pinging him*

*constantly seemed to annoy him”)*

***/(?:P|p)(?:I|i)(?:N|n)(?:G|g)/***

*e) A date in the form of one or two digits, a dot, one or two digits, a dot,*

*two digits*

*(eg. 2.16.13, 03.15.70, 3.5.78)*

***/\d{1,2}(.)\d{1,2}(.)\d{2}/***

*Q2. (15 points) Write a program that reads a string from the standard input, and uses a regular expression to test whether the string is a valid IP address. (eg. 192.62.0.255 is a valid IP address, but 189.9,32 is not)*

***Python Script:***

***Q2\_validate\_ip\_address.py***

*Q3. (15 points) Write a program that reads a number of variable length (eg. 73618, 829, 1, 980), adds up all the digits (would yield 25, 19, 1, 17, respectively), and displays the result.*

***Python Script:***

***Q3\_sum\_digits.py***

*Q4. (20 points) Write a program that simply tokenizes the following text, i.e., by separating the punctuation from the words (,.?!:.....). The program should display the output tokenized. You should try to address special cases such as abbreviations (do not separate the punctuation in an abbreviation, A.B.C., should not yield A . B . C .), apostrophe (should be attached to the letter that follows, e.g., tokenize Peter's as Peter 's), etc.*

*"Predictions suggesting that large changes in weight will accumulate indefinitely in response to small sustained lifestyle modifications rely on the half-century-old 3,500 calorie rule, which equates a weight alteration of 2.2 lb to a 3,500 calories cumulative deficit or increment," write the study authors Dr. Jampolis, Dr. Chaudry, and Prof. Harlen, from N.P.C Clinic in OH. The 3,500- calorie rule "predicts that a person who increases daily energy expenditure by 100 calories by walking 1 mile per day" will lose 50 pounds over five years, the authors say. But the true weight loss is only about 10 pounds if calorie intake doesn't increase, "because changes in mass ... alter the energy requirements of the body’s make-up." "This is a myth, strictly speaking, but the smaller amount of weight loss achieved with small changes is clinically significant and should not be discounted," says Dr. Melina Jampolis, CNN diet and fitness expert.*

***Python Script:***

***Q4\_Tokenize\_String.py***

*Q5. (20 points) Write a program that reads a series of numbers and it displays the digits in ascending order, together with their frequency. The program will also display the number of numbers in the series (Tokens) and the number of unique numbers (Types) in the series*

*E.g., for the input: 4 9 2 7 2 0 2 0 13 23 13. The program will display:*

*0 2*

*2 3*

*4 1*

*7 1*

*9 1*

*13 2*

*23 1*

*Tokens: 11*

*Types: 7*

***Python Script:***

***Q5\_Token\_Frequency.py***

*Q6. (20 points) Repeat Q5 above for the text in Q4 after tokenizing it. I.e. write a program to count the number of strings in the text and sort them alphabetically in ascending order, and also produce the number of tokens and types in the text. (Don’t forget to count the punctuation marks).*

***Python Script:***

***Q6\_Tokenized\_Freq.py***

Q7. (5 points) Ignoring letter case, how many lines of text in the English UNCorpus mention the term Human Rights?

**grep -iwc "Human Rights" uncorpora\_plain\_20090831.tmx**

5664

Q8. (10 points) The Full UNCorpus Answer the following questions using Unix commands and regex only. Each question should be answered with one command line (possibly consisting of multiple piped Unix commands)

a. How many lines does the UNCorpus file have?

**wc -l uncorpora\_plain\_20090831.tmx**

**1501316**

b. How many segments <seg>?

**grep -o '<seg>' uncorpora\_plain\_20090831.tmx | wc -l**

434034

c. How many non-segments? As in tags that are not <seg> like <tuv>?

**grep -E '<([a-z][a-z0-9]\*)\b[^>]\*>.\*</\1>' uncorpora\_plain\_20090831.tmx | wc -l**

488546

**Non <seg> tags = Total tags - #<seg> tags = 488546 - 434034 = 54,512**

d. How many English segments does the text have?

**grep -o '<tuv xml:lang="EN">' uncorpora\_plain\_20090831.tmx | wc -l**

72339

e. How many segments exist for each languages (Chinese, Arabic,...)? (again, done in one command)

**grep -E '<tuv\s\S\*>' uncorpora\_plain\_20090831.tmx | wc -l**

434034

**For each language 434034/6 = 72339**

Q9. (20 points) The English UNCorpus Answer the following questions using Unix commands and regex only. Each question should be answered with one command line (possibly consisting of multiple piped Unix commands) a. Extract the text without XML for only the English segments and put in a file called “uncorpus.eng.txt” (Hint, use “grep –a1”). The rest of the questions are about this file. How would you verify that you did not miss any lines?

**grep -A1 '<tuv xml:lang="EN">' uncorpora\_plain\_20090831.tmx | sed -n 's:.\*<seg>\(.\*\)</seg>.\*:\1:p' > uncorpus.eng.txt**

b. Count the total number of words (tokens).

**wc -w uncorpus.eng.txt**

2685545

c. Count the total number of unique words (types).

**cat uncorpus.eng.txt |tr 'A-Z' 'a-z' | perl -pe 's/\s/\n/g;' | sort | uniq -c |wc**

33365

d. Count the total number of unique words ignoring capitalization

**cat uncorpus.eng.txt |tr " " "\n" | sort | uniq -i| wc**

37021

--- we are also counting the numbers which are not words

e. Count the total number of pure digits tokens.

**grep -E '\d{1}' uncorpus.eng.txt |wc**

48236 2036683 13556224

f. Count the total number of digits with non-word characters with them (e.g.

8,000.00).

**grep -E '\d+(,|.)?\d+' uncorpus.eng.txt | wc**

32939 1401326 9277850

g. Count the total number of words starting with capital letters.

**grep -E '[A-Z][a-zA-Z]\*(\\s+[A-Z][a-zA-Z]\*)\*$' uncorpus.eng.txt | wc -l**

3466

h. What are the top 15 most common first words of sentences

**cut -d “ ” uncorpus.eng.txt –f2 | sort | uniq -c |sort -nr | head -15**

i. What are the top most common (that are not sentence initial).

perl -pe 's/\s/\n/g;' uncorpus.eng.txt | sort | uniq -c| sort -nr | grep -E '[A-Z][a-zA-Z]\*(\\s+[A-Z][a-zA-Z]\*)\*$' | head -10

--- It also selects the single capital character like I M

perl -pe 's/\s/\n/g;' test.txt | sort | uniq -c| sort -nr | grep -E '[A-Z]/S\*' | head -10

j. Count all occurrences of Roman numerals

**grep -E '^M{0,4}(CM|CD|D?C{0,3})(XC|XL|L?X{0,3})(IX|IV|V?I{0,3})$' uncorpus.eng.txt | wc**

759 759 2379

Q10. (10 points) This question uses the file “uncorpus.eng.txt” as the corpus. Compare the four sets of top 10 and the four sets of bottom 10 words. What words are similar, or different? Are you surprised (or not surprised) by the results? (why?)

**cat uncorpus.eng.txt | perl -pe 's/\s/\n/g;' | sort | uniq -c |sort -nr | head -40**

**cat uncorpus.eng.txt | perl -pe 's/\s/\n/g;' | sort | uniq -c |sort -n | head -40**

Q11. (10 points) Back to the Original Corpus a. Get the top 20 (most frequent) words in English, Arabic, Spanish and Russian of the UNCorpus. You will need four separate commands. Show the lists of words in your answer. For this task, consider a word to simply be white-space delimited (i.e. keep all punctuation and digits and separate on white space). b. Use Google Translate to compare the meanings of these words to the English top and bottom words. What words are similar, what are different? Show your work including the results of Google Translate. You can list them in two tables: one for the top words and one for the bottom words.

**grep -A1 '<tuv xml:lang="EN">' uncorpora\_plain\_20090831.tmx | sed -n 's:.\*<seg>\(.\*\)</seg>.\*:\1:p' > uncorpus.eng.txt**

**cat uncorpus.eng.txt | perl -pe 's/\s/\n/g;' | sort | uniq -c |sort -nr | head -20**

267940 the

175497 of

136607 and

99545 to

66802 in

35910 on

32327 for

22534 that

21181 its

20349 with

20126 a

20006 United

19973 as

17251 by

16995 Nations

13933 at

12827 all

12061 international

11986 States

11173 their

**grep -A1 '<tuv xml:lang="AR">' uncorpora\_plain\_20090831.tmx | sed -n 's:.\*<seg>\(.\*\)</seg>.\*:\1:p' > uncorpus.arabic.txt**

**cat uncorpus.arabic.txt | perl -pe 's/\s/\n/g;' | sort | uniq -c |sort -nr | head -20**

92683 في

45238 من

38530 على

34817 -

34466 إلى

24225 أن

18986 التي

18947 وإذ

17776 الأمم

16764 المتحدة

15046 عن

11324 الدول

10979 أو

10175 المؤرخ

9476 كانون

9413 مع

9163 جميع

8916 بما

8800 العام

8770 العامة

**grep -A1 '<tuv xml:lang="ES">' uncorpora\_plain\_20090831.tmx | sed -n 's:.\*<seg>\(.\*\)</seg>.\*:\1:p' > uncorpus.spanish.txt**

**cat uncorpus.spanish.txt | perl -pe 's/\s/\n/g;' | sort | uniq -c |sort -nr | head -20**

313375 de

177307 la

131113 y

93456 en

86469 los

82328 el

77855 las

77003 a

69299 que

52778 del

37328 para

28287 con

22750 su

21639 por

21407 al

20266 sobre

18446 Naciones

15557 se

14062 Estados

13952 Unidas

**grep -A1 '<tuv xml:lang="RU">' uncorpora\_plain\_20090831.tmx | sed -n 's:.\*<seg>\(.\*\)</seg>.\*:\1:p' > uncorpus.russian.txt**

**cat uncorpus.russian.txt | perl -pe 's/\s/\n/g;' | sort | uniq -c |sort -nr | head -20**

135693 и

100876 в

37886 по

37178 на

28031 с

19152 Объединенных

18190 Организации

18148 о

15609 для

14885 от

14180 что

13804 к

13361 Наций

11698 также

10911 года,

9245 декабря

9233 года

8520 их

8379 призывает

7816 или

**Observation:** After translating all the most frequent words of all the above 4 languages it was concluded that in all the languages most of these were stop-words and English & Spanish languages are most similar to each other as the percentage of common words in the most frequent words in these 2 languages was the highest.

I have performed this experiment for to 100 words as well to get to these conclusions.