ASSIGNMENT: 01

README FILE

Methodology:

1.) We have tokenized the input file with RegexpTokenizer ('\w+') for tokenizing the words and for tokenizing the sentence, we have used sent_tokenize.

2.) We are checking the regular for vowels and consonants, We are using **\b[aeiouAEIOU][a-zA-Z]*[0-9]*** regular expression for checking the vowels, i.e a word that is starting from lower case or upper case vowel will be printed. Sample output: ask, Act, aa9.

For consonants we have used **\b[b-df-hj-np-tv-zB-DF-HJ-NP-TV-Z][a-zA-Z]*[0-9]*** regular expression. Sample output: bell, Cat.

3.) We are matching the words with **\S+@\S+[.]\S+** Regular expression to find the email ids present in the file.

4.) We have tokenized the input file with sent_tokenize. Ask the user to give a word as input and print the count of the sentence as well as sentence that are starting from that word using the regular expression **'^'+word+" "**

5.) We have tokenized the input file with sent_tokenize. Ask the user to give a word as input and print the count of the sentence as well as sentence that are ending with that word using the regular expression **" "+word+".?$".**

6.) We have tokenized the input file with sent_tokenize. Ask the user to give a word as input and print the count of the sentence as well as sentence that have the word.

7.) We are matching the content of file with **[a-z A-Z]*[0-9]*\?** Regular expression to find out all the questions present in the file.

8.) We are matching the content of file with **[0123]*[0-9]) [a-zA-Z]* [0-9][0-9][0-9]*[0-9]* (2[0-3]|[01][0-9]):([0-5][0-9]):([0-5]?[0-9])** Regular Expression. Since we are given that we have to consider the time only when dates are given (date followed by time), so first part is checking for dates and **2[0-3]|[01][0-9]):([0-5][0-9]):([0-5]?[0-9]** part is checking for time. Since we have to print only minute and second therefore, we are printing second and third index of tuple.

9.) Abbreviations will have minimum 2 Upper Case letters and the may have lower case letter. We are matching the content of file with **\b(?:[A-Z][a-z]*){2**,} Regular expression to find out all the abbreviations.

Assumptions:

1.) Email-id may contain '@' and '.'.
2.) We are considering case insensitive checking for question 4, 5, 6.
3.) In question 7, we are assuming questions will end with '?' and we are considering case insensitive checking.
4.) Abbreviations will have minimum 2 Upper Case letters and they may have lower case letter

Pre-processing steps:

1.) We are taking file as input and then passing it to the functions.
2.) We have tokenized the file using RegexpTokenizer ('\w+') and sent_tokenize according to the question.
3.) In question 4,5,6 we are converting the whole file and the given input word to lower-case.