CSC 206: NLP **Name (Print):** ____________________
Spring 2015

Problem Set 2: Remembering Python

Due: Friday, March 10 2015, before class

Please create a folder for this problem set, and save all your Python files from Section 2 as well as a pdf file containing the answers to the questions from Section 1 in this folder. Then .zip up the folder and submit the zipped folder on Nexus.

1 NLP Applications

Questions:

1. Call Amtrak at (800) 872-7245 to speak with a dialog system. Find out about some train schedules. Test the system. Try different ways of speaking (e.g. different accents, different intonations, more formal or more colloquial, different words and expressions, ...). What things confuse the system; what works well?

    1. Write 1/2 to 1 page about your experience and what you found out about the capabilities of the system.
    2. Analyze what kinds of knowledge (about language and about the real world) this natural language processing system uses. What additional knowledge would be needed to solve some of the problems that you identified.

2 Some more Python practice

Questions:

2. Write a function `word_lengths` that maps a list of words into a list of integers representing the lengths of the corresponding words.

    Save your program in a file called `word-lengths.py`.

3. Write a function `char_freq` that takes a string and builds a frequency listing of the characters contained in it. Represent the frequency listing as a Python dictionary. For example, `abracadabra` should return the dictionary

    ```
    {'a':5, 'b':2, 'r':2, 'c':1, 'd':1}
    ```

Remember that the entries in a dictionary have no particular order. So the order of the items may be different when your code prints out the dictionary.

Save your program in a file called `frequency.py`.

4. The International Civil Aviation Organization (ICAO) alphabet assigns code words to the letters of the English alphabet acrophonically (Alfa for A, Bravo for B, etc.) so that critical combinations of letters (and numbers) can be pronounced and understood by those who transmit and receive voice messages by radio or telephone regardless of their native language, especially when the safety of navigation or persons is essential. Here is a Python dictionary covering one version of the ICAO alphabet:

```
d = {’a’:’alfa’, ’b’:’bravo’, ’c’:’charlie’, ’d’:’delta’, ’e’:’echo’,
  ’f’:’foxtrot’, ’g’:’golf’, ’h’:’hotel’, ’i’:’india’, ’j’:’juliett’,
  ’k’:’kilo’, ’l’:’lima’, ’m’:’mike’, ’n’:’november’, ’o’:’oscar’,
  ’p’:’papa’, ’q’:’quebec’, ’r’:’romeo’, ’s’:’sierra’, ’t’:’tango’,
  ’u’:’uniform’, ’v’:’victor’, ’w’:’whiskey’, ’x’:’x-ray’,
  ’y’:’yankee’, ’z’:’zulu’}
```

Your task in this exercise is to write a function `speak_ICAO`, which takes any text (i.e. any string) and translates it into ICAO words.

Save your program in a file called `icao.py`.

Make sure that you use the function names and file names that are specified in the instructions.