CMPSC 1500 - Spring 2019

Lab 02

1. A Caesar cypher is a weak form of encryption that involves “rotating” each letter by  
   a fixed number of places. To rotate a letter means to shift it through the alphabet, wrapping around  
   to the beginning if necessary, so ’A’ rotated by 3 is ’D’ and ’Z’ rotated by 1 is ’A’.  
   To rotate a word, rotate each letter by the same amount. For example, “cheer” rotated by 7 is “jolly”  
   and “melon” rotated by -10 is “cubed”. In the movie 2001: A Space Odyssey, the ship computer  
   is called HAL, which is IBM rotated by -1.  
   Write a function called rotate\_word that takes a string and an integer as parameters, and returns  
   a new string that contains the letters from the original string rotated by the given amount.  
   You might want to use the built-in function ord, which converts a character to a numeric code, and  
   chr, which converts numeric codes to characters. Letters of the alphabet are encoded in alphabetical  
   order, so for example:  
   >>> ord('c') - ord('a')

Because 'c' is the two-eth letter of the alphabet. But beware: the numeric codes for upper case  
letters are different.

Also decode the following joke encoded in ROT13 “V'ir urneq vg fnvq gung gur jnl gb cvpx n oneore va n oneorefubc vf gb pubbfr gur bar jvgu gur jbefg unvephg--gurl phg rnpu bguref' unve.”

1. The functions in lower.py are intended to check if a string contains any lowercase letters, but at least some of them are wrong. Describe what each function actually does when the parameter passed is a string.
2. There is a built in tolower() function in python, but write your own version called mytolower() in a file called mytolower.py to see what all is involved. Then run the test bench script *lowerbenchtest.py* and verify your function is correct.

If your function works it just prints ‘True’ for the two test cases. If one of them fails, it prints the failing string from your function. If you run into problems, feel free to duplicate the testbench and modify it however you like to inspect the list returns from tolower(). But your file will be graded by running the lowertestbench.py as provided.

1. Write two scripts to find minmax of 2d array, one using numpy.amax/amin and the other with a for loop.

Create the array with the following line

**Import numpy as np**

**rnd = np.random.rand(1000,1000)**

Pass the array to the script as a parameter so you are sure that each script uses the same array!

Measure the performance of each method by using the time lib and printing the total execution time at the end of the script.