Python Programming (CS0452)
Tutorial Exercises for Unit 3
Dictionaries and Tuples

I. Practice Dictionary - related operations and functions

*II. Two words "alternade" if taking alternating letters from each forms a new word. Write a function 'is_interlocked' that takes a word as input and checks if two or more meaningful words can be formed by this word (subwords need not be of the same length).
For example:
TOTAL = 'schooled', forms ONE = 'shoe' and TWO = 'cold'.
TOTAL ='lacerated', forms ONE = 'let', TWO = 'are' and THREE ='cad'.

* Not mandatory, only a challenge!!

III(a). Write a function word_hist that counts the occurrence of the following words in a given input file 'words.txt': {'the', 'of', 'for', 'is', 'as', 'an'}
Display the words and their count.

III(b). Modify the function in (a) so that the words are sorted.

III(c). Modify the function in (a) so that the words appear in descending order of their frequency.

IV. Consider the input file 'mbox.txt', extract the email-ids that belong to the lines starting with 'From'. Display all the email-ids, also tell which is the most popular mail-id.

V. Consider the input file 'mbox.txt', extract the date and time when the emails were received and list the same (take only the lines that begin with 'From').

VI. Run the 'Memo' version of fibonacci series and the original version. Compare their run times.

VII. Memorize the Ackermann function, check if bigger arguments can be evaluated.

VIII. Given an input text file, 'words.txt', store all the words into a list, count the number of occurrence of each word using dictionary data structure. Display the top 10 most 'popular' words.

IX. Write a function that accepts a multiple line string as argument, counts the number of alphabets, digits, punctuation marks (, . : ; ' ") and spaces. Prepare a dictionary which has the above said elements as keys and their counts as values.

X. Practice Tuples and their related operations

XI. Write a program that accepts a tuple containing numbers as input and outputs two tuples: one containing odd numbers and the other containing even numbers.

[Example:

I/P: (1,5,2,9,8,6,4,2,5)
O/P: even tuple: (2, 8, 6, 4, 2)
odd tuple: (1, 5, 9, 5)]

XII. Consider a tuple containing duplicate elements. Count the number of times each element appears in the tuple. If the appearance is more than once, then print all the indices along with the element.

XIII. Define a function to sort the (name, age, height) tuples in ascending order where name is 'string', age and height are 'numbers'. Start sorting by name, then by age and then by height.