Joanna Klukowska
joannakl@cs.nyu.edu

Assignment 11 (The Last One)
Due Date for Programs: December 13, 2016

Problem 1 (10 points): What does this code do?

Take a look at the code below and the output that it produces. Try to figure out exactly what is going on. Explain what this code does.

```
def extract_alpha ( word  ):
    clean_word = ""
    if len(word) == 0  :
        return clean_word
    if not word[0].isalpha() :
        return clean_word
    for c in word :
        if c.isalpha() or c=='-' or c=='_' or c=='\'' \
                or c==u'\u2019' or c==u'\u2018':  #unicode for smart single quotes
            clean_word = clean_word + c
    if len(clean_word) == 1 and not clean_word.isalpha()  :
        return ""
    return clean_word.lower()

phrase = "You're the guest of the Ford family - the Fords." + \
         "They live on a 4th floor of a three-story brownstone. "  + \
         "Wow!!!"

words = phrase.split()
words_clean = []

for i in range(len(words)) :
    words_clean.append( extract_alpha( words[i] )  )

for i in range(len(words)) :
    print (format(words[i], "<15s"), format(words_clean[i], "<15s") )

```

Output:

```
You're          you're
the             the
guest           guest
of              of
the             the
Ford            ford
family          family
-
the             the
Fords.          fords
They            they
live            live
on              on
a               a
4th
floor           floor
of              of
a               a
three-story     three-story
```

Joanna Klukowska
joannakl@cs.nyu.edu

```
brownstone.     brownstone
Wow!!!          wow
```

Problem 2 (40 points): List Intersection

Write a program that prompts the user to enter two separate lists of numbers (the lists should be arbitrary length and the user input should be terminated by a negative value). Compute the intersection of the two lists (which values occur in both lists) - the program should produce a list of all the values that are in the intersection of those two lists.

Here is a sample run of the program:

Output:

```
Enter values for list 1 (terminate with -1):
    98
    67
    43
    18
    67
    98
    32
    17
    21
    -1

Enter values for list 2 (terminate with -1):
    87
    90
    65
    43
    17
    25
    67
    -1

List 1:
[98, 67, 43, 18, 67, 98, 32, 17, 21]

List 2:
[87, 90, 65, 43, 17, 25, 67]

The lists have 3 elements in common:
[67, 43, 17]
```

Comment your source code by 1) briefly describing parts of your program 2) including your name, the date, and your class section at top of your file (above everything else) 3) **documenting all the functions following the IPO format**

What to Submit

This program should be named (i.e., the name of the file containing the program should be) `list_intersection.py`. You only need to submit the source code for this problem.

Problem 3 (50 points): Word Count

Write a program that once again opens a text of a book. This program should count occurrences of a user specified word in the text. Your program should count exact matches as well as occurrences of that word as a substring in a larger word (for example 'cat' is a substring of 'catfood'). The program should be case-insesitive, i.e., 'cat', 'Cat' and 'CAT' should be counted all as exact matches of 'cat'.

The program should prompt the user for the name of the file containing the book and for the word that he/she wants to search for. The program should produce the results with counts for exact matches and substring matches as well as the total number of words in the input file.

Joanna Klukowska
joannakl@cs.nyu.edu

Finally, the program should print a sorted list of all the UNIQUE words that were matched in the process. For example, if the user entered 'cat' and that produces 10 exact matches as well as 3 matches to 'catfood', 4 matches to 'cats' and one match to 'cataracts', then the list of unique words should contain ['cat', 'cataracts', 'catfood', 'cats'].

NOTE: you will need the code from problem 1 for solving this problem.

Here is a sample run of the program:

Output:

```
Enter the file name: moby_dick.txt
Enter the word to search for: whale

--------------------------------------------------------------------------------

total number of words:  215823
whale occurs 967 times by itself
whale occurs 739 additional times as a substring of a larger word

--------------------------------------------------------------------------------

Here are the unique words:

fishersright-whale  horse-whales        jonas-in-the-whale  narwhale
narwhalehowever     narwhales           right-whale         sperm-whale
sperm-whalemen      whale               whale-balls         whale-boat
whale-boats         whale-boat's        whale-bone          whale-bones
whale-books         whale-craft         whale-cruisers      whale-cry
whale-e             whale-fastener      whale-fish          whale-fishers
whale-fishery       whale-fleet         whale-ground        whale-hater
whale-hunt          whale-hunter        whale-hunters       whale-hunting
whale-jets          whale-killer        whale-lance         whale-line
whale-lines         whale-naturalists   whale-pike          whale-pole
whale-ports         whale-ship          whale-ships         whale-ships'
whale-ship's        whale-smitten       whale-spades        whale-spout
whale-steak         whale-surgeon       whale-teeth         whale-trover
whale-wise          whalea              whaleanother        whaleas
whaleboats          whaleboat's         whalebone           whaleboning
whaled              whaledid            whaledrive          whaleeven
whaleho             whalehow            whalein             whaleman
whaleman's          whalemen            whalemento          whalemen's
whalemoby           whalemodifying      whaleno             whaler
whalers             whales              whalesa             whaleship
whaleships          whaleshirr          whalesmen           whalesnow
whalesquid          whalethe            whalethis           whale'
whale's             whale'sno
```

Comment your source code by 1) briefly describing parts of your program 2) including your name, the date, and your class section at top of your file (above everything else) 3) **documenting all the functions following the IPO format**

What to Submit

This program should be named (i.e., the name of the file containing the program should be) `word_counter.py`. You only need to submit the source code for this problem.

What and how to submit?

You should submit the source code file for each program to NYU Classes by the due date stated above. Make sure that you get an email confirmation after you submit the assignment. You should keep that email until the grades are returned - it is your proof that the assignment was submitted! If you do not get an email confirmation, you should try to resubmit the assignment. If you do not get that email, it means that we did not get your assignment.

Joanna Klukowska
joannakl@cs.nyu.edu

Name(s) and NetId(s):

What does this code do?

Describe in details what the function does to a `word` that is passed to it:

Describe what happens to the `phrase` variable: