Lab 8 - Functions

Part 1: Code some functions for the lab from chapter 8 (lab 7)

Take the program you wrote (the word/letter count) and write functions to do some of the work. In particular, write functions that:

1. Remove the punctuation from the included string
2. Create the count dictionaries. Code one function that may be used to create both word and letter count dictionaries; one dictionary for each function invocation (don't have ONE function create BOTH dictionaries with ONE invocation)
3. Print out the word and letter count tables. Again, ONE function, TWO invocations.

The outputs should be identical to those from the previous word/letter count lab from Chapter 8

Part 2: Code a series of functions that print the first N elements of a structure

Write the following functions that print N elements of a structure (collection):

```
1. def print_first_n_set_values( set_tb_printed,
   num_to_print ):
2. def print_first_n_list_values( list_tb_printed,
   num_to_print ):
3. def print_first_n_dict_values(dict_tb_printed,
   num_to_print):
```

The above functions may be coded with *comprehensions* but not required (but **highly desired!)**

Copyright 2018 © Man on Fire, Inc. All rights reserved

Write **another function** that calls one of the above 3 functions *based on the type of the structure (list, set, dictionary)*

```
def print_first_n_values( structure_tb_printed, num_to_print
<with additional code> ):
```

The above function with that "<with additional code>" is what you'll code to make *num_to_print* **an optional parameter**. Code the rest of *num_to_print* to **use a default value of 'all'.** If *the user does not pass a second arg to print_n_first_values,* **print the entire structure** (need a way for print_first_n_values to pass a number that is the *size of the structure* to the appropriate function (1-3 on the previous page)

Here's a sample run:

```
First 3 LIST values:
  [1, 2, 'a']

First 3 SET values:
  {1, 2, 'll'}

First 3 DICTIONARY values:
  {1: 'a', 2: 'BB', 3: 'cc'}
<class 'tuple'> has no special print function

Using default argument for # items to print

First 6 LIST values:
  [1, 2, 'a', 'b', 4.5, '5']
First 5 SET values:
  {1, 2, 'll', 'd', 23}
First 4 DICTIONARY values:
  {1: 'a', 2: 'BB', 3: 'cc', 'X': 'Some text'}
<class 'tuple'> has no special print function
```


Copyright 2018 © Man on Fire, Inc. All rights reserved

Given these definitions:

```
a_list = [1, 2, 'a', 'b', 4.5, '5']
a_set = {1, 2, 'd', 'll', 23}
a_dict = { 1: "a", 2: "BB", 3: "cc", "X": "Some text"}
a_tuple = (1, 'a', 3)
```

Note the **tuple has no special function**

Two solutions are provided: one uses if/elif/else logic; the other uses a *dictionary of functions keyed on the data type.*

Use whatever technique you like.

For the highly ambitious (optional):

Create a module named *print_n_structure_elements* and import the module into a program that has the above structures coded with the function calls.

Copyright 2018 © Man on Fire, Inc. All rights reserved