CSC 401 ASSIGNMENT SIX

Due Date: Tuesday, Oct. 24th by 11:58 PM

The purpose of this assignment is to assess your understanding of

- Dictionary
- Tuple
- Set
- Random Module

SUBMISSION

- Include your full name as a comment on the first line of your Python program file.
- Include the problem number as a comment before each user defined function.
- Code all the problems in one Python file(.py) labeled as YourName_HW6.py
- Upload one file to Submissions folder.

PROBLEM

Note: You may not use Python statements, functions, data type, etc. that were not discussed in the reading assignment or the lecture notes/video for this week or previous weeks. This is a class for students who have not programmed before and I expect everyone to code on the same level. If you have a better way of writing the code, then upload two versions: one that codes according to the specifications and the other that demonstrates advanced programming techniques.

The assignment covers material introduced in weeks six and seven. In this assignment the solution to each problem should be coded as a user-defined function. Be sure to include a return statement in each function even if nothing is returned. All of the functions should be in one file.

1. (Indexing, 20 pts) At the end of books, there usually is an index that lists the pages where a certain word appears. In this problem, you will create an index for a text but, instead of the page number, you will use the line numbers. You are to implement the function **index (fname, letter)** that takes as input the name of the text file and the first letter of the words for which you are to create the line number index. For each word that begins with 'letter', you are to print the corresponding line numbers. You should

    a. open and read the file only once; use readline()

    b. Remove all punctuation from the text

    c. For each word that begins with the specified letter, find all the line numbers in which the letter appears.

    d. Print the total number of lines in the text and the number of words that begin with the specified letter.

Use a **dictionary** to store results **{word: list of line number(s)}.** For each word, a line number should be listed only once, for example, 'the' appears 3 time on line 250; 250 should not be listed 3 times, only once. In the output, the line numbers should be separated by commas, but they are not printed as lists (no brackets). The words in the output should be in alphabetical order, i.e. a – z.

```
>>> index('Pride_and_Prejudice.txt','k')
Word        Line Nbr.
keener      2159
keenest     983, 2665, 2939
keep        103, 205, 247, 261, 311, 485, 647, 653, 667, 669, 919, 957, 1417, 1825, 2027, 2473, 2523, 2893, 2977, 3465, 3567,
3579, 3641, 3685, 4231
keeping     1463, 1553, 3023, 3933
keeps       1491, 2307, 2849
kept        927, 997, 1495, 1545, 1849, 1865, 1905, 2457, 2517, 2813, 3073, 3525, 3583, 3595, 3861, 4069, 4241
kill        3323
killed      2971, 3051, 3493, 3569
kind        215, 223, 443, 545, 775, 937, 949, 977, 1033, 1103, 1167, 1255, 1305, 1357, 1491, 1511, 1547, 1637, 1663, 1753, 1
767, 1865, 2071, 2107, 2119, 2129, 2153, 2181, 2293, 2327, 2395, 2445, 2503, 2677, 2733, 2805, 2881, 2935, 2971, 2977, 2989,
2993, 3039, 3069, 3081, 3199, 3217, 3257, 3367, 3397, 3807, 3859, 3889, 3937, 4027, 4145, 4215, 4253
kinder      2187, 4225
kindest     2849
kindled     3575
kindly      927, 1259, 1457, 1687, 1861, 2159, 2287
kindness    117, 137, 269, 467, 469, 601, 665, 1071, 1337, 1491, 1705, 1737, 1755, 2035, 2239, 2243, 2257, 2301, 2325, 2337,
2339, 2353, 3201, 3221, 3241, 3613, 3711, 3985, 4199
kindred     3957
kinds       2393
king        221
kingdom     163, 1287, 1801, 3273
kiss        2945
kissed      3461, 3735
kitchen     957
knees       2855
knew        123, 133, 209, 371, 585, 767, 961, 1017, 1067, 1129, 1143, 1837, 1865, 1977, 2067, 2219, 2235, 2241, 2267, 2391,
2449, 2695, 2707, 2711, 2771, 2831, 2863, 2875, 2895, 2927, 2981, 3001, 3011, 3049, 3065, 3217, 3259, 3367, 3385, 3461, 3535
, 3547, 3567, 3613, 3741, 3763, 3929, 3989, 3999, 4019, 4077, 4177, 4183, 4213, 4225
knighthood 221, 1753
know        31, 37, 49, 65, 77, 93, 111, 117, 121, 137, 161, 175, 199, 203, 217, 231, 235, 275, 287, 289, 293, 299, 309, 385,
445, 565, 609, 625, 645, 705, 733, 753, 775, 789, 795, 833, 843, 899, 953, 957, 987, 1001, 1011, 1035, 1059, 1111, 1115, 112
1, 1125, 1127, 1153, 1195, 1255, 1263, 1265, 1267, 1345, 1351, 1353, 1369, 1371, 1417, 1453, 1455, 1469, 1477, 1523, 1531, 1
597, 1605, 1639, 1667, 1699, 1703, 1723, 1737, 1765, 1767, 1781, 1791, 1833, 1967, 1975, 1981, 1983, 2001, 2013, 2015, 2079,
2091, 2093, 2095, 2115, 2117, 2145, 2177, 2199, 2229, 2241, 2245, 2247, 2285, 2303, 2325, 2355, 2391, 2425, 2427, 2435, 2437
, 2465, 2561, 2593, 2631, 2639, 2659, 2669, 2713, 2721, 2725, 2731, 2775, 2791, 2799, 2837, 2845, 2849, 2861, 2871, 2881, 29
15, 2919, 2923, 2925, 2927, 2929, 2971, 2975, 2977, 2995, 3007, 3023, 3033, 3035, 3045, 3059, 3123, 3143, 3149, 3183, 3201,
3215, 3223, 3243, 3261, 3299, 3331, 3333, 3337, 3349, 3363, 3367, 3381, 3397, 3409, 3419, 3459, 3471, 3475, 3491, 3493, 3497
, 3499, 3515, 3517, 3519, 3529, 3561, 3565, 3575, 3597, 3661, 3693, 3713, 3825, 3833, 3849, 3851, 3895, 3903, 3913, 3933, 39
43, 3947, 4001, 4007, 4049, 4081, 4085, 4087, 4103, 4105, 4109, 4125, 4141, 4143, 4145, 4201
knowing     567, 579, 953, 1181, 1229, 1783, 1857, 1985, 2129, 2253, 2255, 2311, 2457, 2555, 2557, 2629, 2781, 2847, 2997, 30
01, 3013, 3247, 3281, 3661, 3861, 3997
knowledge   89, 313, 385, 575, 1535, 1839, 1841, 1925, 1983, 2067, 2233, 2239, 2243, 2435, 2461, 2931, 3257, 3263, 4075, 4259
known       21, 129, 281, 287, 1017, 1049, 1125, 1441, 1443, 1491, 1627, 1639, 1683, 1753, 1899, 1935, 2095, 2125, 2213, 2235
, 2257, 2259, 2273, 2511, 2519, 2523, 2527, 2529, 2591, 2663, 2697, 2729, 2787, 2789, 2791, 2817, 2823, 2849, 2863, 2875, 30
01, 3073, 3083, 3207, 3365, 3373, 3535, 3825, 3971, 3985, 4069, 4077, 4241
knows       1155, 1737, 2195, 2663, 2719, 2923, 3023, 3491, 3513

There are 4272 lines in the book.
There are 29 words that begin with " k "
```

2. (Random, 15 pts) Craps is a dice-based game played in many casinos. The game starts with the player throwing a pair of standard, six-sided dice. If the player rolls a total of 7 or 11, the player wins. If the player rolls a total of 2, 3 or 12, the player loses. For all other roll values, the player will repeatedly roll the pair of dice until either she rolls the initial value again (in which case she wins) or 7 (in which case she loses). Write a function **craps()** that takes no arguments, simulates one game of craps, and returns 'I win' if the player won and 'I lost' if the player lost. Consider implementing a second function if the player does not win or lose on the initial roll.

```
>>> craps()
6
3
6
'I win'
>>> craps()
7
'I won'
>>> craps()
9
8
6
7
'I lose'
```

3. (dictionary, 15 points) Implement a function **STUDENT ()** which allows the user to enter the 7-digit student ID. The program will keep prompting the user for a last name and first name. If the student does not have a student ID on record, the program will then ask for the student ID, and store that information. If the student already has a student ID, the program will display it, and ask for confirmation whether a new studentID should be assigned (and, if so allows the new studentID to be entered). When the user hits the return key, the program prints a report listing all students with their studentID. Your report does not have to be in alphabetical order, however, if you do want to make it alphabetical, you can use sorted on the keys of your dictionary.

```
>>> studentID()
First name: Harry
Last name: Potter
Student ID: 1234567
First name: Ron
Last name: Weasley
Student ID: 2345678
First name: Mickey
Last name: Mouse
Student ID: 4567890
First name: Harry
Last name: Potter
('Potter', 'Harry') has ID 1234567 Update? y
Student ID: 4839201
First name: Mickey
Last name: Mouse
('Mouse', 'Mickey') has ID 4567890 Update? n
First name:

Contents of dictionary:
Mouse, Mickey has studentID 4567890
Potter, Harry has studentID 4839201
Weasley, Ron has studentID 2345678
```

IF YOU HAVE ANY QUESTIONS REGARDING THIS ASSIGNMENT, PLEASE POST THEM TO THE ASSIGNMENT SIX DISCUSSION FORUM.