CIS 122 Project 5 Read text files, count words using a dictionary
Winter 2015

Project 5

5a 10 points

Write a function **make_word_list** that is given a line of text to process and returns a list of words in the line.

Example

```
line = "Romeo! I'm tired--aren't you? Take me to the 'Primavera' restaurant."
```

returns a word_list like this, all lower case.

```
['romeo', "i'm", 'tired', "aren't", 'you', 'take', 'me', 'to' , 'the', primavera, 'restaurant']
```

Notice that you want to strip ' quote marks from the beginning and ending of words, but not from the interior of words like "aren't".

Hint: copy the line character by character, copying alphabetic characters, blanks and ' quote marks. Replace all other characters with blanks. Split the newly copied line on blanks.

Test your function with the line of text above and print the resulting word list, word by word

```
romeo
i'm
tired
aren't
you
take
me
to
the
primavera
restaurant
```

5b 10 points

Read the **short.txt** file using this url

http://www.cs.uoregon.edu/Classes/15W/cis122/data/short.txt

Use your `make_word_list( )` function from 5a to break the text up into words.

Create a **word_count dictionary** from the words in the word lists.

Print the entire little dictionary, **sorted** into word order.

5c 15 points

Change the url to

http://www.cs.uoregon.edu/Classes/15W/cis122/data/romeo_and_juliet.txt

Add a **while loop** to ask for a **word to look up**.

Show the **word** and the **count** of how often it was used.

Your dialog might look somewhat like this:

```
Type word to find, or X to exit: thou
thou..................... 277

Type word to find, or X to exit: juliet
juliet................... 65

Type word to find, or X to exit: methinks
methinks................. 3

Type word to find, or X to exit: x

Done
```

5d 10 points

Use turtle graphics to draw a "random walk". Each "step" will be 10 pixels long.

Hint

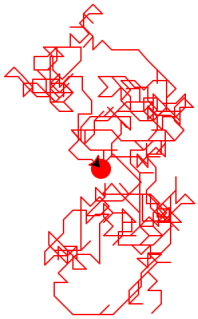
```
import random
import turtle as t   # lets you do t.forward(x) for example
```

You'll find it helpful to define 2 functions

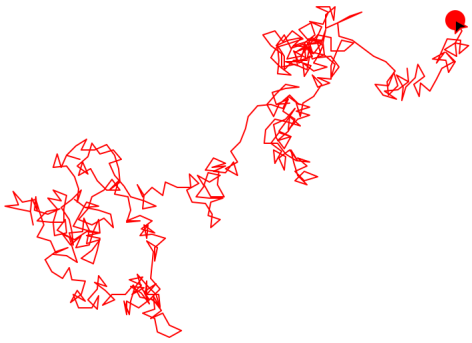
```
def jump(distance):
    ''' Move turtle distance units without leaving a mark'''

def random_walk(n_steps):
    ''' Starting the turtle at home,
        choose a random angle to turn, then
        move the turtle forward a few steps in that direction
        repeat n_steps times.
   '''
```

Sample outputs - notice the dot at the end of its walk

Chooses one of 8 angles

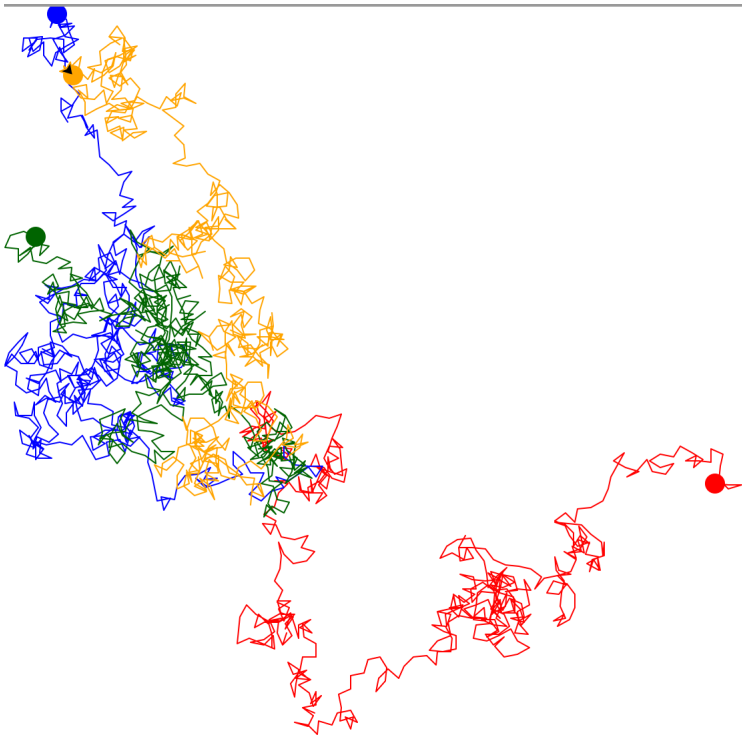


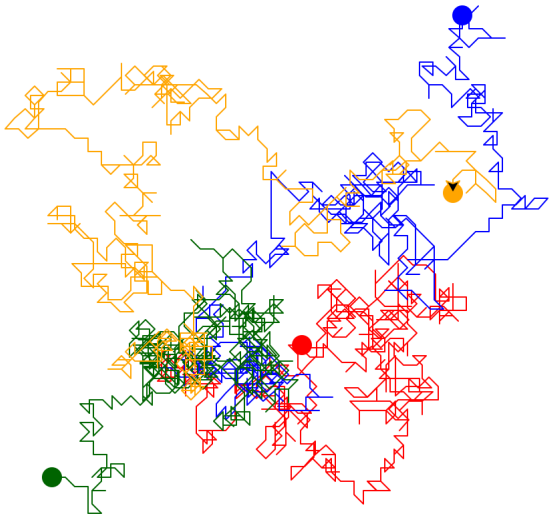
Chooses one of 360 angles



5e 5 points

Produce **3 or 4 random walks**, each starting from the "home" (0,0) location; and each using **a different color**.





XC **5 points**

Modify 5c to "**invert" your dictionary:***
Create a dictionary whose key is count (such as 65, and whose value is a list of words that occur that many times somewhat like this tiny sample shows:

```
{65 : ['juliet'], 277 : ['thou', 'love']}
```

See section 11.4 Inverting a dictionary in your Practical Programming text for a worked example.

** When you invert your dictionary, you don't change the original dictionary, you just produce another (inverted) dictionary.*

Due *Friday March 13, 6 PM.*

Accepted with a **5 point late penalty** *by Monday, March 16 6 PM.*