CIS 122 Fall 2015 Project 6
Due Wednesday, November 18, 11:59 PM

Briefly

Submit several Python 3 programs
Your programs are worth a total of 25 points

"**http://www.cs.uoregon.edu/Classes/15F/cis122/data/**xxx.txt"

where you must change xxx.txt to the correct name of the text file

Notice that the url is **case-sensitive** after .edu
Classes, not classes,
15F not 15f
cis122 not CIS122
data not Data not DATA

Test your programs -- did they work right? -- before uploading to Canvas.

P6_states.py 10 points:
Learning objective
Read a text file from the web
`http://www.cs.uoregon.edu/Classes/15F/cis122/data/`
`state_data.txt`

Control the print based on changes in the data stored in your list

Read a text file (**state_data.txt**) from the web to print a list of items. It has state, capital, area in square kilometers, population in millions such as 4.6.

Print each item you read on a separate line. Print a total of the area and population for the states your user has selected, as in this example.

Example

```
Your choice: s
State names starts with: a

State          Capital          Square KM   Millions
Alabama        Montgomery         135,765        4.8
Alaska         Juneau           1,717,854        0.7
Arizona        Phoenix            295,254        6.6
Arkansas       Little Rock        137,732        1.0

Total                           2,286,605       13.1
```

Rubric 10 points total

5 point **Reads** each line from the web page, stripping off \n and "decoding" the line as "utf-8", then splitting the line on commas into a list with
- state
- capital
- sq km (int)
- population (float)

2 points **Appends** state little_list of 4 items to a **states_list**

1 Iterates (works through) each state in the state list

2 **Prints** item_list from states_list
At end of printing data for the selected states,
prints total area in sq km
prints total population (in millions)

P5-scrabble.py 10 points:
Read a text file (sowpods.txt) from the web,
`http://www.cs.uoregon.edu/Classes/15F/cis122/data/`
`sowpods_short.txt`

After putting the data into a list (note: no sub-list needed here) repeatedly ask user to type a word, then either announce that the word is in the list, or that it's not found.

One working, change to the full file **`sowpods.txt`** with over 260,000 words.
`http://www.cs.uoregon.edu/Classes/15F/cis122/data/`
`sowpods.txt`

Learning objectives
Gather data from a web page.
Store it in a list.
Interact repeatedly with a user to check whether a word is in the list.
Gain confidence in the capacity of your computer and Python to handle fairly large amounts of data.

```
print("Yes", word, "is in the list")
```
or
```
print("Sorry", word, "is in not in the list")
```

Rubric
1 point Gets the data from a web page
1 point Tested with **sowpods_short.txt**, then uses the full size **sowpods.txt** file.

1 point Reads each line, stripping off **newline** character
1 point **Decodes** the line using **utf-8**

1 point **Skips** any line starting with **#**
1 point Stores the word into a **word list**

2 points while **loop** asks
1) **L** look for word to search for or
Q Quit
2) Asks for word to search for

2 points **prints that word is in the list** or
word is not found in the list.

Note: this file is the official world-wide (except US, Canada and Thailand) Scrabble ® word list. It has some unusual words but no "proper nouns" like Thomas or Armenia.

P6_Circles.py 5 points

Learning obective

Some **fun** with **colorful graphics** and **random** numbers.

Using random.**randrange**(small,limit) instead of randint(small, large)

Note: be sure to import both random and turtle such as this way:

```
import random
import turtle as t
```

`random.randint(1,6)` returns numbers 1, 2, 3, 4, 5, or 6

But if you have a list of colors such as

```
color_list = ['red', 'blue', 'yellow' ]
```

you want to get random numbers like 0, 1, 2 -- numbers that start at 0 and go up to but not including the length of the color_list.

```
n = len(color_list)
index = random.randrange(0, n)
```

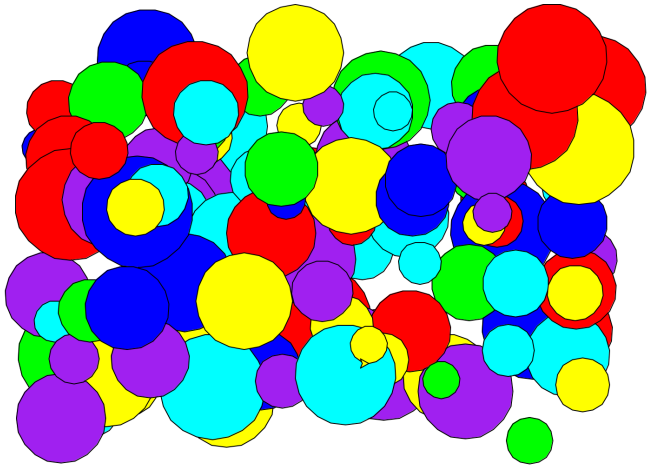
gives you 0, 1, or 2 in this example.

So use random.randrange to choose items at random from a list.

You might want to define a function such as
`random_one_of(a_list)` which will return an item from the list, picked at random.

random.randint is a good choice for numbers such as the x and y location to draw at.

Create a series of at least 60 circles with radius varying from around 20 to 60, and colors selected at random from a list of at least 6 distinct colors.



Rubric

1 point	import both turtle and random
1 point	create a loop that runs 60 to 300 times
1 point	the turtle jumps to a random x,y position without drawing a line
2 points	the turtle draws a colored circle of random radius filled at random with a color from a list of colors