NAME:
EMAIL:
SIGNATURE:

Lehman College, CUNY
CMP 108/MAT 135/SOC 251: Programming for Data Analysis
Final Exam– Version 1
Spring 2017

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What does the Python code draw:

```
import turtle
tess = turtle.Turtle()
for i in range(5):
     tess.forward(100)
     tess.left(72)
```

Output:

2. What will the following Python code print:

```
myFriends = "Linus Torvalds,Steve Jobs,Bill Gates,Monty Python"
print(myFriends[-1])
friends_list = myFriends.split(",")
count = len(friends_list)
print("I have", count, "good friends:")
for f in friends_list:
    print(f)
favorite = friends_list[0].split(" ")
print("My favorite friend is", favorite[1])
favorite = friends_list[0].split(" ")
what = favorite[0].replace("s","x")
print("who invented", what.upper())
```

Output:

3. Write a **complete** Python program that implements the pseudocode:

   (a) Ask the user for the number of days until finals.

   (b) Print out the weeks until finals.

   (c) Print out the leftover days until finals.

4. (a) Write a **complete** Python program that prompts the user for a file name and prints the number of spaces in the file.

(b) Write a **complete** Python program that prints the **minimum 2010 population** stored in a data file. Your program should open the file, `population.csv` and sum the last values in each line. The data is separated by commas ("`,`"). Your program should print the total sum that you calculated.

population.csv:

```
Borough, 2000 Population, 2010 Population
Bronx, 1332650, 1385108
Brooklyn, 2465326, 2504700
Manhattan, 1537195, 1585873
Queens, 2229379, 2230722
Staten Island, 443728, 468730
```

5. Fill in the missing function definitions for this Python program:

```
def main():
    welcome()             #Prints "Welcome" to the screen
    age = userInput()     #Continues to prompt until user enters a positive
                          #number and returns that number
    y = calculate(age)    #Using age, calculates year born
    displayResults(age,y) #Prints age and birth year
main()
```

(That is, write the functions `welcome()`, `userInput()`, `calculate()` and `displayResults()`.)

6. (a) Write a Python function that takes number between 1 and 5 as a parameter and returns the corresponding number as a string. For example, if the parameter is 1, your function should return `"one"`. If the parameter is 2, your function should `"two"`, etc. If the parameter is not between 1 and 5, your function should return the empty string.

(b) Write the Python code for the function below:

```
getInput()
   Ask user for a string
   Until they enter a non-empty string
      Print error message
      Ask user for a string
   Return the string entered
```

7. The file `cunyLocations.csv` contains the locations of the City University of New York. The first couple of lines of the file are:

```
College or Institution Type,Campus,CampusAddress,City,State,Zip,Latitude,Longitude
Senior Colleges,Baruch College,151 East 25th Street,New York,NY,10010-2313,40.740977,-73.984252
Senior Colleges,Brooklyn College,2900 Bedford Avenue,Brooklyn,NY,11210-2850,40.630276,-73.955545
Community Colleges,Borough of Manhattan Community College,199 Chambers Street,New York,NY,10007-1044,40.717367,-74.012178
```

Fill in the missing lines of code for the Python program to plot the CUNY colleges, color coded by type:

```
import folium
from folium.plugins import MarkerCluster
import pandas as pd

#a)   Read in the CSV file into a pandas dataframe:




#b)  Set up a folium map centered at [40.75, -74.125]



coords = []
popups = []
icons = []
for index,row in cuny.iterrows():
    lat = row["Latitude"]
    lon = row["Longitude"]

    #c)  Extract the campus name:



    coords.append([lat,lon])
    popups.append(name)

    #d)  If the college is a senior college, make the icon blue.  Otherwise, make a green icon








#e)  Save the map to 'cunyLocations.html':
```

8. What will the following R code print:

Output:

```
> sentence <- c(’walk’, ’the’, ’plank’)
> sentence[3]
> sentence[3] <- "dog"
> sentence[4] <- "home"
> sentence[c(1, 3, 4)]
> a <- c(1, 2, 3)
> a + 1
> a / 2
> a - a
> select <- a == c(1, 1, 1)
> a[select]
```

9. Fill the R code that will do the following:

(a) Create a vector, `temps`, of the high temperatures in New York City in March 2017:

```
>                      (66,63,38,29,35,44,50,59,60,47,28,30,35,32,27,39,47,38,47,
                        51,59,49,43,55,56,42,50,46,58,51,43)
```

(b) Create a vector, `rainFall`, of the rain fall recorded in New York City in March 2017:

```
>                      (0.12, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.10, 0.02,
                        0.00, 0.31, 0.00, 0.00, 0.00, 1.97, 0.00, 0.00, 0.00, 0.08,
                        0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.01, 0.14, 0.26, 0.72,
                        0.00, 0.01, 1.51)
```

(c) Print the total rain fall recorded for the month:

```
>
```

(d) Compute the correlation of temperature to rain fall for month’s data:

```
>
```

(e) Make a plot of temperature versus rain fall:

```
>
```

10. Write a R program that will plot the built-in data set `cars` (car speed and stopping distances recorded in the 1920s. The structure of the data set is:

```
str(cars)
'data.frame': 50 obs. of  2 variables:
 $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
 $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
```

Your plot should display:

- the speed versus the stopping distance,
- the x-axis should be labeled: "Speed",
- the y-axis should be labeled: "Stopping Distance",
- the plot should be labeled: "Cars Data from 1920's", and
- display the average stopping value as an `abline`.