NAME:
EMAIL:
SIGNATURE:
CIRCLE COURSE SECTION: TTH 11-1 TTH 6-8

Lehman College, CUNY
CIS 166 Final Exam, Version 1, Spring 2015

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What will the following code print:

```
s = "Ada=>Lovelace=>Charles=>Babbage"
a = s[0:3]
print(a.upper())
names = s.split("=>")
print(names)
b,c,d = names[1],names[2],names[3]
print(c,d)
print(b[-1]+"n"+d[-2]+"ine")
print('Put_line: ("', a.lower(),'")')
```

Output:

2. Write a **complete program** to calculate how much something will weigh on Mars. Your program should prompt the user for the weight on the Earth and then print out the weight on Mars. For example, if the user enters `100`, your program should print out `38`.

   The weight of an item on Mars is 38% *of its weight on earth.*

3. What is output of the code below:

```
def prob4(fred, george):
     if fred < 2:
          print("Small case")
          harry = -1
     else:
          print("Complex case")
          harry = helper(fred,george)
     return(harry)
```

```
def helper(isaac,jacob):
     s = ""
     for j in range(isaac):
          print(j, ": ", jacob[j])
          if j % 2 == 0:
               s = s + jacob[j]
               print("Building s:", s)
     return(s)
```

(a)
```
r = prob4(0,"herbert")
print("Return:  ", r)
```

Output:

(b)
```
r = prob4(2,"lehman")
print("Return:  ", r)
```

Output:

(c)
```
r = prob4(4,"college")
print("Return:  ", r)
```

Output:

4. Given the following program and input file, what is printed:

```
def prob5V1():
   c = 0
   infile=open("places.txt","r")
   for line in infile.readlines():
      if len(line) > 7:
         print("Long Line: ", end ="")
         c = c + 1
      print(line)
   print("Num long lines is", c)

prob5V1()
```

places.txt

```
Vandenberg
Wright-Patterson
Laughlin
Dover
Charleston
San Antonio
```

Output:

5. (a) Write a function that takes number between 1 and 7 as a parameter and returns the corresponding day of the week as a string. For example, if the parameter is 1, your function should return `"Monday"`. If the parameter is 2, your function should `"Tuesday"`, etc. If the parameter is not between 1 and 7, your function should return the empty string.

(b) Write a `main()` that allows the user to enter a number and calls your function to show that it works.

6. Complete the following program, which sets up a graphics window and turtle, draws a hexagon (6-sided figure) to the window, and then prints a closing message and closes the graphics window when mouse is clicked. That is, write the functions `setUp()`, `drawHexagon()`, and `conclusion()`:

```
import turtle

def main():
    w,t = setUp()   #sets up a graphics window and turtle
    drawHexagon(t)  #draws a hexagon using the turtle
    conclusion(w)   #prints goodbye and closes window on click

main()
```

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

(b) Write a **complete** program that prints the total population stored in a data file. Your program should open the file, `population.csv` and sum the last values in each line. Note that the first line should not be used since it contains the column headers and not data. 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
```

8. Write the Python code for the algorithms below:

(a)
```
getInput()
Ask user for an even number
Until they enter an even number
    Print error message
    Ask user for an even number
Return the even number entered
```

(b)
```
merge(ls, mid)
Initialize the variables: set newList to be an empty list, set counters i to be 0
and j to be mid.
While i < mid and j < len(ls):
    If ls[i] < ls[j], then append ls[i] to the newList and increment i.
    Else: append ls[j] to the newList and increment j.
While i < mid:
    Append ls[i] to the newList and increment i.
While j < len(ls)
    Append ls[j] to the newList and increment j.
Return newList
```

9. In lab, we wrote a Tic-Tac-Toe program. Change the program to check for a winner after each move and keep track of the number of times this occurs. Your program should print out a message if someone has a winning configuration, print out the total winning configurations seen so far, and continue playing.

   Clearly mark your changes to the design below:

```
#Second Version of Tic-Tac-Toe
from turtle import *
def setUp():
    win, tic = Screen(), Turtle()
    tic.speed(10)
    win.setworldcoordinates(-0.5,-0.5,3.5, 3.5)
    for i in range(1,3):
        tic.up()
        tic.goto(0,i)
        tic.down()
        tic.forward(3)
    tic.left(90)
    for i in range(1,3):
        tic.up()
        tic.goto(i,0)
        tic.down()
        tic.forward(3)
    tic.up()
    board = [["","",""],["","",""],["","",""]]
    return(win,tic,board)
def playGame(tic,board):
    for i in range(4):
        x,y = eval(input("Enter x, y coordinates for X's move: "))
        tic.goto(x+.25,y+.25)
        tic.write("X",font=('Arial', 90, 'normal'))
        board[x][y] = "X"
        x,y = eval(input("Enter x, y coordinates for O's move: "))
        tic.goto(x+.25,y+.25)
        tic.write("O",font=('Arial', 90, 'normal'))
        board[x][y] = "O"
    x,y = eval(input("Enter x, y coordinates for X's move: "))
    tic.goto(x+.25,y+.25)
    tic.write("X",font=('Arial', 90, 'normal'))
    board[x][y] = "X"
def checkWinner(board):
    for x in range(3):
        if board[x][0] != "" and (board[x][0] == board[x][1] == board[x][2]):
            return(board[x][0])  #we have a non-empty row that's identical
    for y in range(3):
        if board[0][y] != "" and (board[0][y] == board[1][y] == board[2][y]):
            return(board[0][y])  #we have a non-empty column that's identical
    if board[0][0] != "" and (board[0][0] == board[1][1] == board[2][2]):
        return(board[0][0])
    if board[2][0] != "" and (board[2][0] == board[1][1] == board[2][0]):
        return(board[2][0])
    return("No winner")
def main():
    win,tic,board = setUp()   #Set up the window and game board
    playGame(tic,board)       #Ask the user for the moves and display
    print("\nThe winner is", checkWinner(board))  #Check for winner
```

10. (a) In lab, we processed name data maintained by the Social Security Administration. Write a **function** that takes as input a string of Social Security Administration name data and returns the gender (`'F'` for female and `'M'` for male).

Here are some sample lines from the `NY.txt` file containing the data for New York State:

NY.txt

```
NYM1910Herbert,83
NYM1910Leo,80
NYM1910Andrew,79
NYM1910Ernest,79
NYM1910Milton,79
```

(b) Write a **complete program** that uses your function above to count the number of entries by gender. Your program should open the file `NY.txt`, count the number of lines for males and the number of lines for females, and print the results.

Useful String Methods: (from p 140 of textbook)

Function	Meaning
s.capitalize()	Copy of s with only the first character capitalized.
s.center(width)	Copy of s is centered in a field of given width.
s.count(sub)	Count the number of occurrences of sub in s.
s.find(sub)	Find the first position where sub occurs in s.
s.join(list)	Concatenate list into a string using s as a separator.
s.ljust(width)	Like center, but s is left-justified.
s.lower()	Copy of s with all characters converted to lowercase.
s.lstrip()	Copy of s with leading whitespace removed.
s.replace(oldsub,newsub)	Replace all occurrences of oldsub in s with newsub.
s.rfind(sub)	Like find, but returns rightmost position.
s.rjust(sub)	Like center, but s is right-justified.
s.rstrip()	Copy of s with trailing whitespace removed.
s.split()	Split s into a list of substrings.
s.title()	Copy of s with first character of each word capitalized.
s.upper()	Copy of s with all characters converted to uppercase.

Useful Turtle Methods: (from http://docs.python.org/3.0/library/turtle.html)

Function	Meaning
t.forward(d)	Move turtle forward d steps
t.backward(d)	Move turtle backward d steps
t.right(angle)	Turn turtle angle degrees to the right
t.left(angle)	Turn turtle angle degrees to the left
t.up()	Pull the pen up: no drawing when moving
t.down()	Pull the pen down: drawing when moving
t.color(c)	Change pen color to color c
t.goto(x,y)	Move turtle to coordinates (x,y)
w.bgcolor(c)	Change background color to color c
w.setworldcoordinates(x1,y1,x2,y2)	Resize drawing area with lower left corner as (x1,y1) and upper right corner (x2,y2)
w.exitonclick()	Closes graphics window on mouse click