CSCE 4430 ASSIGNMENT #4

Due Wednesday, November 16, 2016

1. (Command line Arguments) Write a Python script named ***count.py*** that takes command line arguments. The first argument will be a filename. The second argument will be a word. Your Python script should return the number of occurrences regardless of casing. (20 points)
   ```
   test.txt => 123 123 abc
   python count.py test.txt 123 => 2
   python count.py test.txt abc => 1
   python count.py => ERROR: Not Enough Arguments
   ```

2. (Matrices and File I/O) Write a Python script named ***cluster_counter.py*** that takes command line arguments. The first argument will be a filename. The file loaded will contain an n by m matrix of different characters. This n by m matrix denotes different clusters where each same character represents a group. Casing is not important (i.e ***A*** and ***a*** are the same group). Touching characters (meaning adjacent and diagonal) of the same group are assumed to be in the same cluster. Write a program that tells the user how many groups exist and for each group, how many sub clusters exist. (20 points)
   Example file *test.m*
   ```
   A B A A A
   A B A A A
   B B A A A
   ```
   ```
   python cluster_counter.py test.m
   You have 2 groups: A, B
   The Number of clusters are listed below
   A : 2
   B : 1
   ```

3. (Classes) Write a Python script named ***employee_stats.py*** that takes in command line arguments. The first argument will be a filename. The second argument will be a number corresponding to the results that will be generated. The file loaded will contain employee information in a csv format(***employee.csv***). I strongly recommend you use Python classes for each employee. The print out for each number for the second argument is below. (60 points):
   1. List all the employees by last name in ascending order.
   2. Get the average salary of all the employees for all weeks
   3. Print the names and EPID of the employees with the highest salary for Entry, Middle, and Senior Positions.
      ```
      python employee_stats.py employee.csv 1
      Abbot
      Arango
      …
      ```

Extra Points: 5 points:
John accidently created an ***employee.html*** file and not an ***employee.csv*** file. Employee.html has the data in a table. Make it so employee_stats.py can read data from a csv or parse it from a html file.