CS410P

LAB 13L

Purpose

The purpose of this assignment is to give you practice with reading from records from an input file, separating out the items in each record and writing to an output file.

Problem

There is a data file called UN.csv available in the public folder contains the names and some details of each country that belongs to the UN. Each line of the input file contains the record of one country – each record contains 4 items separated by a comma as the delimiter. The items in each record are in this order:

- Country
- Continent
- Population in millions
- Area in square miles

Our goal is to to create two new files which have all the records but just containing the population and country per record (separated by a comma delimiter). The first output file should order the countries by total population (also display the 10 most populous countries with their population and name on the screen) and the second file should order the countries by population by square mile. Both files should be sorted by population numbers in descending order.

Sample of the input file:

```
Afghanistan,Asia,31.8,251772
Albania,Europe,3.0,11100
Algeria,Africa,38.3,919595
Andorra,Europe,.085,181
Angola,Africa,19.1,481354
Antigua and Barbuda,North America,.091,108
Argentina,South America,44.0,1068302
Armenia,Asia,3.1,11506
Australia,Australia/Oceania,22.5,2967909
Austria,Europe,8.2,32383
Azerbaijan,Asia,9.7,33436
Bahamas,North America,.32,5358
Bahrain,Asia,1.3,253
Bangladesh,Asia,166.3,55599
Barbados,North America,.29,167
Belarus,Europe,9.6,80155
```

Output file Samples

First Output File	Second Output File
1355700000.0,China 1236300000.0,India 318900000.0,United States 253600000.0,Indonesia 202700000.0,Brazil 196200000.0,Pakistan 177200000.0,Nigeria 166300000.0,Bangladesh 142500000.0,Russian Federation 127100000.0,Japan 120300000.0,Mexico 107700000.0,Philippines 96600000.0,Ethiopia 93400000.0,Vietnam 86900000.0,Egypt 81600000.0,Turkey 81000000,Germany 80800000.0,Iran	40789.47,Monaco 20423.05,Singapore 14186.05,Solomon Islands 10243.9,The Republic of Macedonia 5138.34,Bahrain 3391.3,Maldives 3388.43,Malta 2991.06,Bangladesh 1736.53,Barbados 1651.84,Mauritius 1462.21,Lebanon 1404.26,San Marino 1268.71,Republic of Korea 1209.56,Rwanda 1187.5,Nauru 1100.0,Tuvalu

Details

- We want to read the contents of the data file one line at a time. We can use the "split" function to separate out individual items. Remember that each line of the file also has a "carriage return" or "newline" character which needs to be stripped first and then the "split" function applied to create a list with the individual items.
  Here's an example of a string whose individual items are separated by a delimiter which can be split and the resulting values go into a list.

```
sample_string =  "Martin,10/7/95,NH,03061"
list_of_items = sample_string.split(",")
print(list_of_items)
```

Will display:

```
['Martin', '10/7/95', 'NH', '03061']
```

Keep in mind that this is only a suggested approach. If you want to use the CSV library in Python you are welcome to do so.

Suggested approach:

1) The objective is to create a new list of tuples where each tuple contains a pair of values (population and country). As every record is read, add a tuple of 2 items to this new list until all the data is read.
2) Next you sort the list in descending order (by default sorting a list happens in ascending order – if you use the argument (reverse=True) with the sort method it will sort the list in descending oder.
3) Next we want to display the top 10 most populous countries (on the screen)
4) We also want to create a new text file (call it by any name you wish, to be created in the same folder as the program) and write the contents of the new list into this file. When this is done your output file should look exactly like my output file
5) Keep in mind that we need two types of outputs (both screen display and output file) – second one showing population density per square mile (in descending order).

- It's up to you to create any functions that you think would be good to make your program modular with functions created as necessary to separate out different operations, along with

separating out code into multiple files. Any repetitive code should ideally be placed in a function. 10 points are set aside for program design.

Input

You need to prompt the user for the input file name. The input file consists of a series of records, each record has four fields/items separate by commas.

Output

The output files must be created as described above – a series of records, each record containing two fields/items separated by a comma. Also display a screen output of the 10 most populous countries with their population and name. Sample **screen output** is shown below:

```
Input File Name:UN.csv
Most Populous Countries
     Population                        Country
  1355700000.00                          China
  1236300000.00                          India
   318900000.00                  United States
   253600000.00                      Indonesia
   202700000.00                         Brazil
   196200000.00                       Pakistan
   177200000.00                        Nigeria
   166300000.00                     Bangladesh
   142500000.00             Russian Federation
   127100000.00                          Japan

Most Populous Countries by Area
     Population                        Country
       40789.47                         Monaco
       20423.05                      Singapore
       14186.05                Solomon Islands
       10243.90      The Republic of Macedonia
        5138.34                        Bahrain
        3391.30                       Maldives
        3388.43                          Malta
        2991.06                     Bangladesh
        1736.53                       Barbados
        1651.84                      Mauritius
```

Grade Key

A	Comments (including Name, brief description about program)	**5**
B	Input data read correctly from file (input data file name prompted)	**15**
C	Two lists created successfully containing relevant information, sorted correctly	**20**
D	Screen display conforms to specification	**20**
E	Output files created correctly with data in the correct format	**30**
F	Program design with good programming practices	**10**