CIS 41A - Lab 6: advanced use of functions, generators, review: regex, OOP, containers

Write an application that lets the user look up colleges that are top feeder schools to large tech companies.

**Overview**

Lab 6 consists of 2 files that are the backend and frontend of the application.

colleges.py: backend that reads in data from an input file and provides methods to search the data.

ui.py: frontend that interacts with the user to let user look up colleges and display the search result.

**Input file**

The app uses the file *colleges.txt*, which is the html text of the table named “Top Feeder Rankings (adjusted for undergraduate enrollment)” of the web page <https://www.collegetransitions.com/dataverse/top-feeders-tech>

- The rows of the table that contain data for a college have the following html format in the *colleges.txt* file  
 <td>1</td><td>Carnegie Mellon University</td><td>1,356</td><td>Google</td><td>Dropbox</td> </tr>

- The 3 green data need to be extracted from the row:

* school name
* top company by number of graduates (company 1)
* top company by ratio of graduates / total employees (company 2)

- Rows that don’t have college data are not in the above format.

**colleges.py**

There are 2 classes.  
A class to represents one college, which has:

1. A method to initialize the college object with 3 attributes: name, company 1, company 2.
2. Other methods as needed.

A class that contains college objects, which has:

1. A method to create college objects from the lines in the input file and store the college objects.
2. A method that returns colleges by ranking:
   * Return a *generator* of college objects, in the same ranking order as in the input file.
3. A method that returns colleges by company:
   * Accept one or more company names as input arguments (*individual names*, not a list / tuple / set of company names).
   * For each input company, find all college names with company 1 as the input company and all college names with company 2 as the input company. Store all the different college name groupings in a container to return.
4. A method that returns all unique company names.
5. Additional method as needed.
6. Note that the backend and does not interact with the user.

**ui.py**

A class to interact with the user.

1. A method to initialize a backend object and print the total number of colleges.
2. A method to let the user view colleges by ranking, one college at a time:

* Print: the college name, company 1, company 2
* Then wait for the user to: - press the Enter key to print the next college info  
   or - press any other key to end the printing
* If the user keeps pressing the Enter key until there’s no more colleges, print an ending message. The code should not need to use the total number of colleges.
* See sample output.

1. A method to let the user view colleges by company name:

* Print all the company names in sorted order, comma separated, in 2 – 3 lines of output.
* Ask the user for company names, separated by comma. There could be 0, 1, or more spaces in between each company name and comma. The company name is case insensitive.
* Call the backend object’s method to access colleges by company name (you should only need to call this method one time).
* For each company name, print all colleges where the company is company 1, and all colleges where the company is company 2, or print “no colleges”
* Print 2 or 3 schools per line, comma separated, to shorten the output. See sample output.

1. A run method that coordinates all the interactions with the user:

* Print a menu with 3 choices: r. Show schools by ranking

c. Show schools by company

q. Quit

* Read the user choice (case insensitive) and process the user choice until the user chooses ‘q’.
* Instead of using an if else statement to process the user choice, create a look up table to quickly determine the method to process the user choice and to print an error message if the choice is not valid.

1. Other methods as needed.
2. At the end ui.py, use the following code to start the UI object:  
    UI().run()

**Additional requirements**

* Handle exceptions, such as file open error, as needed.
* Put your name at the top of each file
* Have a docstring for all public methods

**Sample output** (user input is in blue)

30 top colleges at tech companies # from initializing the backend object

r. Show schools by ranking

c. Show schools by company

q. Quit

Enter choice: 4 # invalid choice

r, c, or q only

r. Show schools by ranking

c. Show schools by company

q. Quit

Enter choice: R # case insensitive

Printing one school at a time

After each school, press Enter to continue, any other key to stop

1 Carnegie Mellon University: Google, Dropbox

2 Columbia University: Google, LinkedIn

a # user ended printing

r. Show schools by ranking

c. Show schools by company

q. Quit

Enter choice: r

Printing one school at a time

After each school, press Enter to continue, any other key to stop

1 Carnegie Mellon University: Google, Dropbox

2 Columbia University: Google, LinkedIn

3 Stanford University: Google, Airbnb

< the schools in between are removed from this sample output to shorten it >

29 University of Waterloo : Google, Stripe

30 University of Michigan: Google, Meta

End of college list # print ending message when there’s no more college

r. Show schools by ranking

c. Show schools by company

q. Quit

Enter choice: c

Here are the companies: # print sorted company names, comma separated, on 2 lines

Airbnb, DocuSign, Dropbox, Google, HubSpot, LinkedIn,

Meta, Microsoft, Snap, Stripe, Twilio, Twitter

Enter company names, separated by comma: Airbnb , meta,abc # various whitespaces, case insensitive

No school with Airbnb as first company

Schools with Airbnb as second company:

Stanford University, California Institute of Technology,

Rice University, Santa Clara University

No school with meta as first company

Schools with meta as second company:

Harvard University, Swarthmore College,

Yale University, Johns Hopkins University,

University of Michigan

No school with abc as first company

No school with abc as second company

r. Show schools by ranking

c. Show schools by company

q. Quit

Enter choice: q