Computer Science 1 — CSci 1100
Lab 5 — Yelp Data
Fall Semester 2018

Lab Overview

This lab explores the use of lists and logic to analyze real data provided by Yelp for restaurants near Rensselaer. You will use a module that we provide to help you parse a file of data. It contains file functions that we have not yet covered in class, but don't worry, we will cover this material very soon. In the meantime, feel free to look at the code and ask about what it is doing during the lab.

To get started, create a folder in your Dropbox for Lab 5 and download the zip file `yelp.zip` from the Piazza Resources page into this new folder. Unzip it to extract two files, `lab05_util.py` and `yelp.txt`. The first is a module for reading the second file. Use the Wing IDE to take a look at `lab05_util.py`. You can see it has functions for parsing the `yelp.txt` file, but no code to actually call these functions.

Checkpoint 1

Let's use the `lab05_util.py` module to read `yelp.txt` into a list and do some simple operations on it. Create a new file called `check1.py` in the same folder as the files from the zip folder and include the following code inside:

```
import lab05_util

restaurants = lab05_util.read_yelp('yelp.txt')
```

Use a few print function calls to test the file read and see the contents of the list. For example, to see the first element, use: `print(restaurants[0])`, you will get:

```
["Meka's Lounge", 42.74, -73.69, \
 '407 River Street+Troy, NY 12180',\
 'http://www.yelp.com/biz/mekas-lounge-troy', \
 'Bars', \
 [5, 2, 4, 4, 3, 4, 5]]
```

Printing other elements will give you similar information for a different restaurant in the list. So:

- The variable `restaurants` contains a list.
- Each element of the list is information on a specific restaurant, also organized as a list. This restaurant information contains (in order): name, latitude and longitude, street address, URL, the type of restaurant and a list of scores given by Yelp users.
- Importantly, the last element of the list of restaurant information is another list, with multiple scores given by Yelp users.

Your job in the first checkpoint to develop a `print_info` function to print information for a single restaurant. Here is the result of printing the first two restaurants in the list followed by the 26^{th} element of the list:

```
Meka's Lounge (Bars)
        407 River Street
        Troy, NY 12180
Average Score: 3.86

Tosca Grille (American (New))
        200 Broadway
        Troy, NY 12180
Average Score: 2.50

Troy Market (Farmers Market)
        Uncle Sam Atrium
        49 4th St
        Troy, NY 12181
Average score: 4.62
```

The first line in the output is the name of the restaurant, followed by its restaurant type in parentheses. The second, third, and possibly fourth or more lines are the address components. Typically this is the street address and the city/state/zip code; but sometimes there is more information and more fields. All address fields should be printed after first printing a TAB character. The final line is the average score, obtained by taking the average of the last entry in the restaurant. You may assume there is at least one score, and you must output the average score accurate to two decimal places.

The one unknown is how do we split the address into multiple lines? Think about our handy friend `split()` that splits a string into a list based on a given delimiter. For example:

```
>>> title = "My,name,is,red"
>>> title.split(",")
['My', 'name', 'is', 'red']

>>> title = "My|name|is|red"
>>> title.split("|")
['My', 'name', 'is', 'red']
```

You can use the split function on the restaurant address to get the pieces you need.

To get you started, here is a basic organization for printing just the name of a restaurant.

```
import lab05_util

def print_info(restaurant):
    print(restaurant[0])

######## main code starts here
restaurants = lab05_util.read_yelp('yelp.txt')
print_info(restaurants[0])
```

To complete Checkpoint 1, show the lab TA or a mentor the code and the output for

restaurants 0, 4 and 42.

Checkpoint 2

Before getting started, copy `check1.py` to `check2.py` and develop your code for `Checkpoint 2` in `check2.py`.

Now modify your code to ask the user for the id of a restaurant from 1 up to and including the length of the `restaurants` list (humans don't need to know about list ids starting at 0). Assume the user enters a number. If the user enters a value outside of the value range, print a warning and do nothing else.

If the user has entered a valid id, you will print the information for the restaurant corresponding to this id (remember: id 1 corresponds to list index 0). Test your code well first to make sure that you only print a restaurant for a valid id.

The second task in this part is to improve on the print function by changing the average score computation. Given the scores for a restaurant, you will drop the max and the min, and find the average of the rest. Remember that you do not have to explicitly remove the max, and min from the list, just subtract their values from the sum.

Once you have computed the average, instead of printing it, print one of the following based on the score:

Score	Output
0 up to 2	This restaurant is rated bad, based on x reviews.
2 up to 3	This restaurant is rated average, based on x reviews.
3 up to 4	This restaurant is rated above average, based on x reviews.
4 up to 5	This restaurant is rated very good, based on x reviews.

where x is the total number of reviews for this restaurant.

Beware: it does not make sense to remove max and min if there are less than three reviews for a restaurant. In this case, we should use the average of all the values (another if statement!).

To complete Checkpoint 2, show the lab TA or a mentor the code and the output. Please check to make sure your code follows the structure we require: first imports, then functions and then the actual code. Test your code with values 8, 22, 33, and 44.

Checkpoint 3

Please come to lab for the last checkpoint.