Checkpoint 1

Let's use the given module to read this file into a list. 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 lab03_util restaurants = lab03_util.read_yelp('yelp.txt') Use a few print statements to see the contents of the list. As the list is large, let's look at the first element: print restaurants[0] We will get: ["Mekas 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]] The variable restaurants contains a list. Each element of this list corresponds to a specific restaurant. From the above example, we see that the first element of the restaurants list is also a list. The information provided for each restaurant is: name, latitude, longitude, street address, URL, the restaurant category, and a list of scores given by Yelp users. Yes, the last element here is yet another list!

Your job in the first checkpoint is to print information for a single restaurant by writing a function called print_info(). Below is the result of printing the first two restaurants in the list: Mekas 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 The first line shows the name and the category in parentheses. The second and third lines both come from the address (note the use of the TAB here). The final line is the average score, obtained by taking the average of the last entry in the restaurant. How do we split the address into two lines? There is a very useful function called split() that splits a string into a list based on a given delimiter. For example: >>> title = "The,Old,Man,and,the,Sea" >>> title.split(",") ['The', 'Old', 'Man', 'and', 'the', 'Sea'] >>> title = "The!Old!Man!and!the!Sea" >>> title.split("!") ['The', 'Old', 'Man', 'and', 'the', 'Sea'] To get you started, here is the basic organization for printing just the name of a restaurant. import lab03_util def print_info(restaurant): print restaurant[0] ####### main code starts here restaurants = lab03_util.read_yelp('yelp.txt') print_info(restaurants[0]) To complete Checkpoint 1, show a mentor the code and the output.

Checkpoint 2

Copy check1.py to check2.py and continue to work on check2.py. Modify your code to ask the user for the id of a restaurant between 1 and 155 (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 range 1-155, print a warning and do nothing else.

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

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

Given this average of the remaining scores, 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. Note that x is the real number of reviews for this restaurant that are used in calculating the average. Beware! It does not make sense to remove max and min if there are less than three reviews for a restaurant. In that case, we should use the average of all the values (another if statement!).

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

Checkpoint 3

Copy check2.py to check3.py and continue to work on check3.py. We will add a final flair in this part to your program from part 2.

Your program should work exactly as it did in part 2. After printing the restaurant info, ask the user the following: What would you like to do next? 1. Visit the homepage 2. Show on Google Maps 3. Show directions to this restaurant Your choice (1-3)? ==> For all of these options, using formatted strings will really simplify your life!

If the user answers 1, then open a browser window using the following command (but use the URL for the business instead of this address). Remember to import module webbrowser first of course. webbrowser.open('http://xkcd.com/1319/') If the user answers 2, then open a browser window with Google maps showing the address of the business. webbrowser.open('http://www.google.com/maps/place/business-address-goes-here') If the user answers 3, then open a browser window with Google maps with the address of the business and Rensselaer. Here is an example call: webbrowser.open('http://www.google.com/maps/dir/business-address/rpi-address') For example, to find the location of Rensselaer, you can use the following call: webbrowser.open(http://www.google.com/maps/place/110 8th Street Troy NY 12180)

Luckily, Google can handle spaces or even pluses in an address.

If the user answers anything else, your program does nothing.