**Edgar Escobedo**

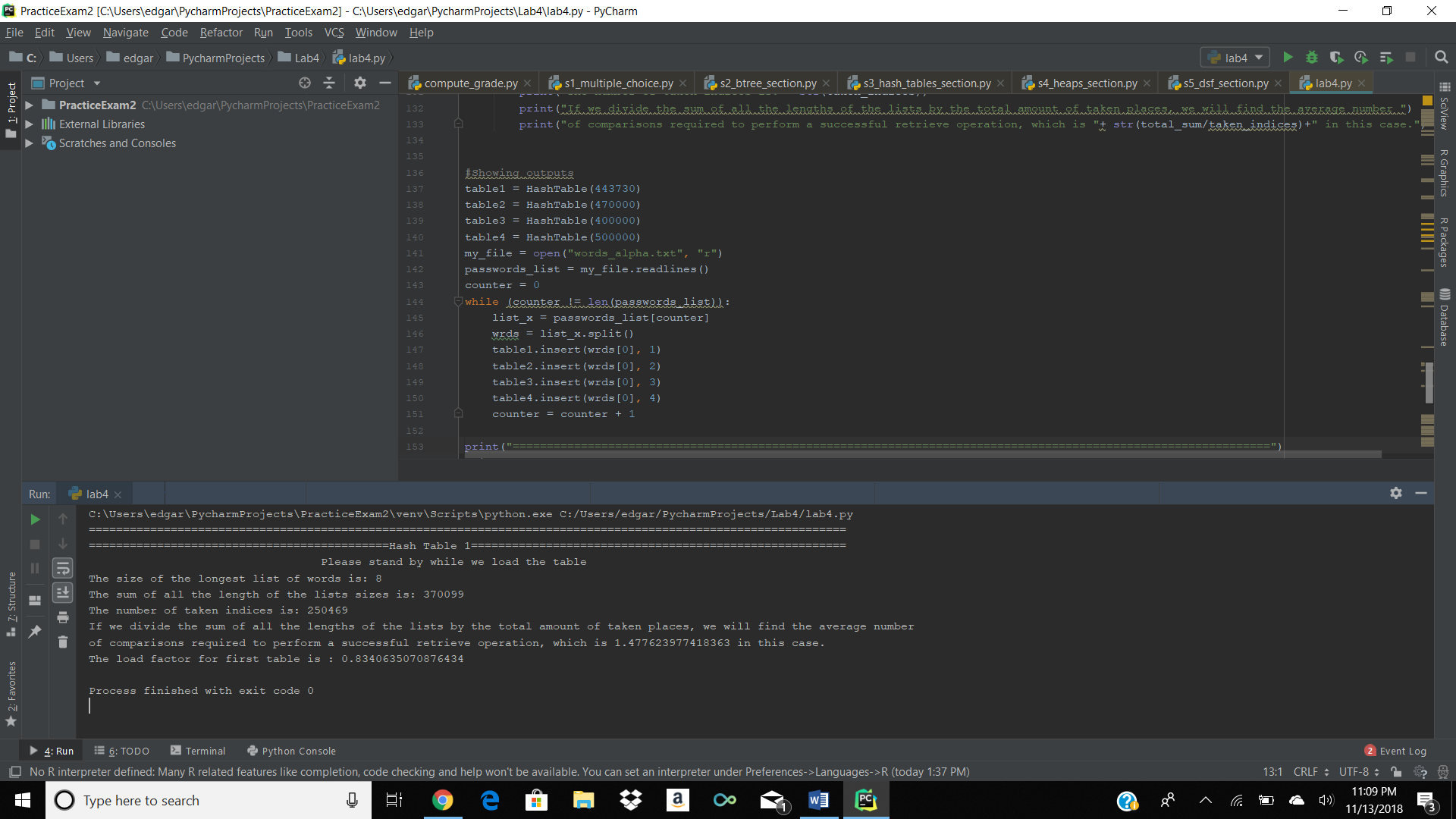
**Lab 4 Report**

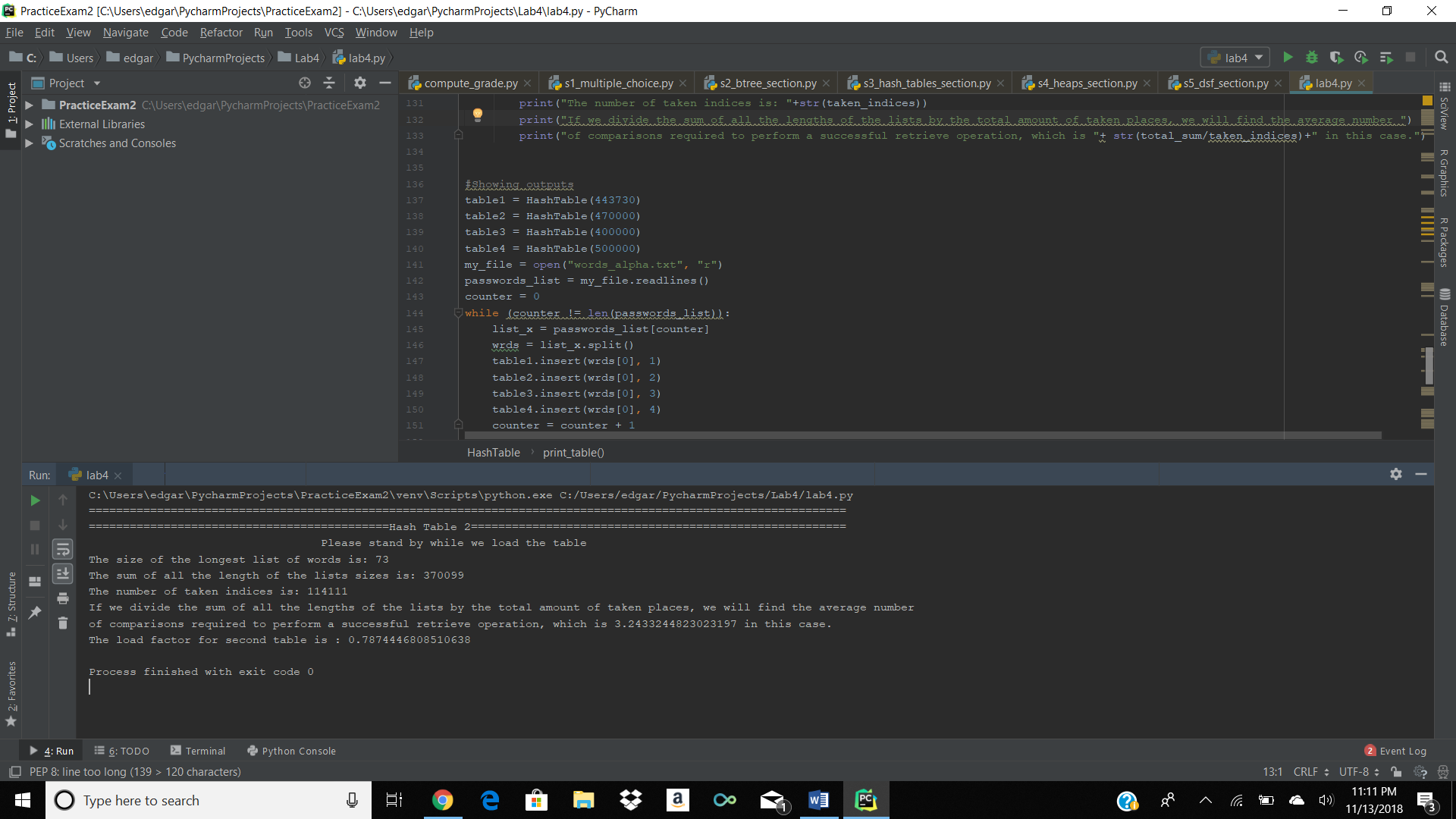
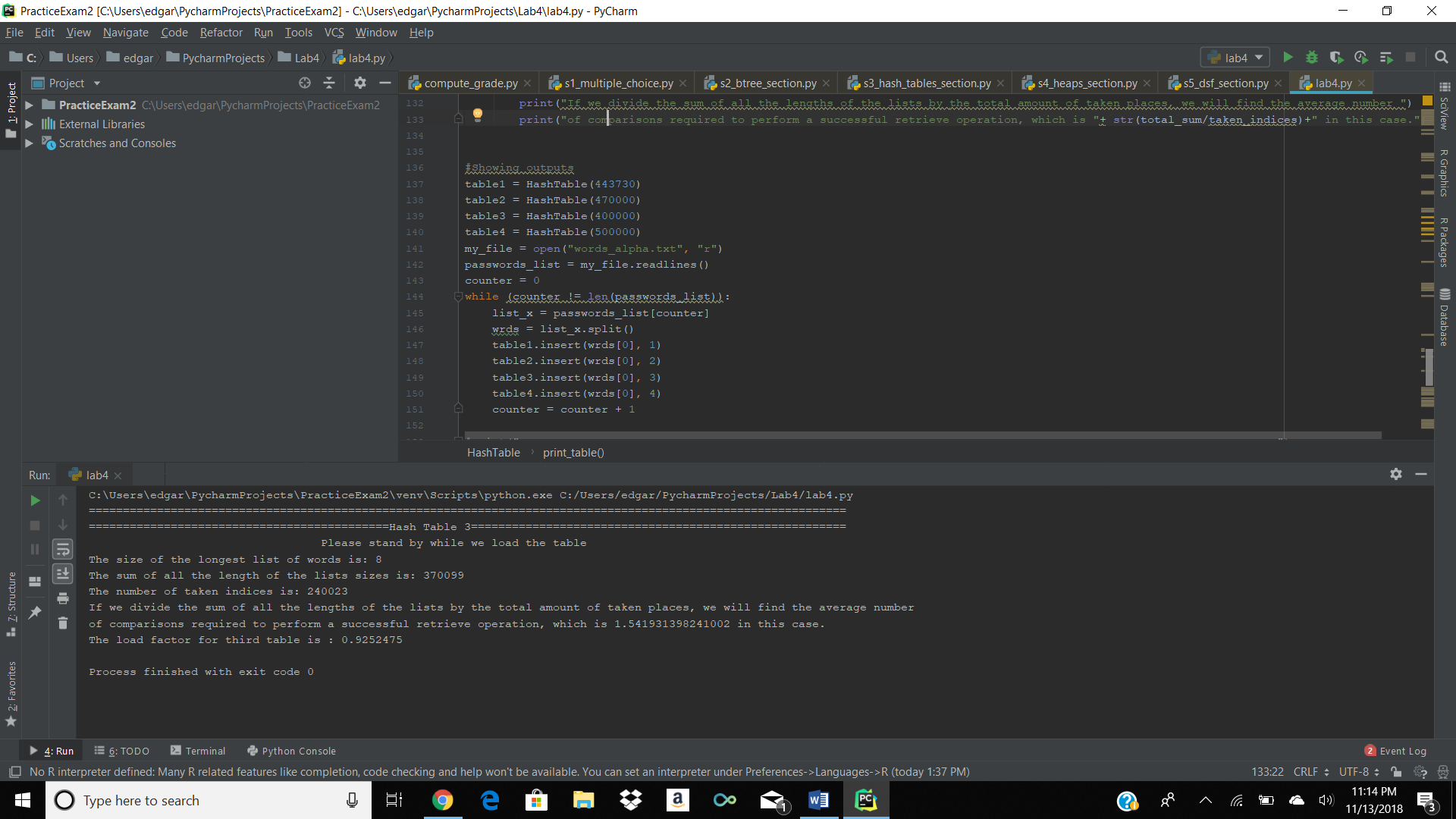
**CS 2302**

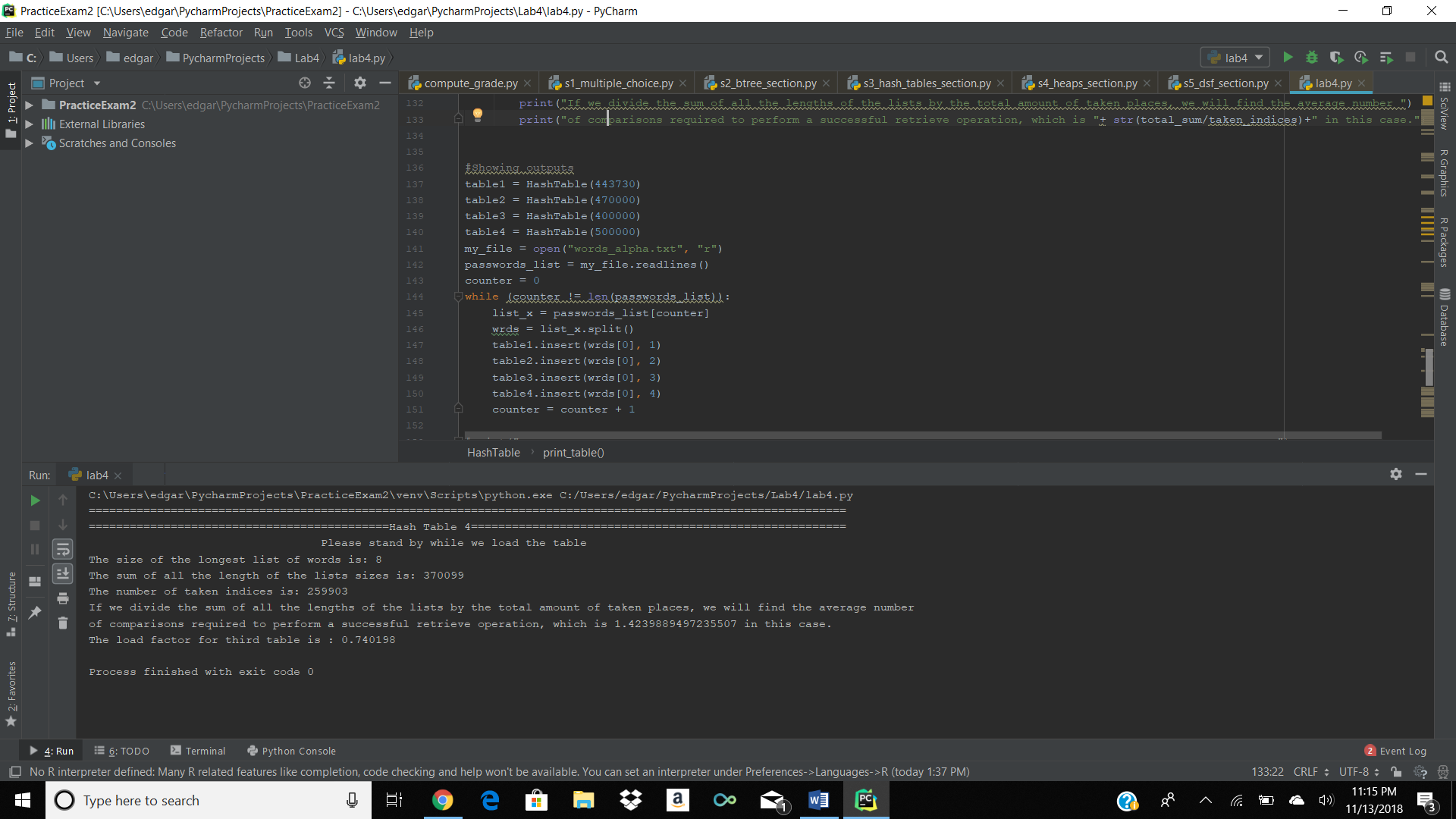
**Professor Aguirre**

As a continuation of lab 3, lab 4 required us to again store the all the English letters that would be given in a text file and store them in a data structure, being a hash table for this lab. In order to do this lab, there were some considerations that had to be accounted for. We had to decide the way that we were going to create the hash function, given that we were given a string (the English word) instead of a number. Then after having the hash function giving us an index, we needed to find the load factor of this hash table, which could be found by the used indexes and the size of the table. Finally, we were asked to give the average comparisons that our program had to make to find a desired value.

In order to create the desired program, I started by reading the file, and separating each word using the split method. Then before inserting the words I decided to create 4 tables with different load factors (meaning the sizes of the hash tables were not the same). When I had each word, I had the option to send each of the words to 3 different hash functions (hash table 3 and 4 shared the same hash function) in order to better show how the hash functions affected the average comparisons each table had. These results and comparisons will be given at the end of the explanation. When the insertions were completed, I created a method which would get the length of the largest list in the current hash table and use that number to create a list that would store all the lists with that length, which would give me the total number of words in the hash table. Using the previously created list, I would delete the list of length 0 (meaning empty indices) and it would give the total indices used during the insertion. Then using the total number of English words and dividing it by the used indices that would give me the average number of comparisons that would change depending on my hash tables.

 To better explain this part of the code, some outputs would be shown.





As a brief description of what the captures mean, the 4 captures represent 4 different hash tables, where table 1 has a unique hash function table 2 has a unique hash function and table 3 and 4 share the same hash function. As seen from the first screenshot, hash table 1, even when it doesn’t have the greatest size, has a very good average comparisons value because of the good hash function that was used to create it. The second screenshot shows a “larger than the first table”, table, which has a not particularly good hash function, so even when it has a smaller load factor than the first table, the hash function plays a crucial role for the average comparisons. Finally, we have table 3 and 4, which as mentioned before share the same hash function, only difference is that table 3 has a greater load factor than table 4. As seen from the average comparisons the load factor also plays a role in the average comparisons value, because of the smaller value for table 4, where the only difference was the smaller load factor.

In conclusion again, this was a very interesting lab, where I could more deeply learn not only the main idea of a data structure, but the logic behind it. The required search for a good hash function and how that function would affect the number of comparisons my hash table would have was extremely educative. This lab up to today has been my favorite lab because of the fun behind it, hash tables has been one of my favorite data structures and having more time dedicated to learning more about them is always appreciated for me.

**Appendix:**

[**https://github.com/ejescobedo/Lab-4/blob/master/lab4.py**](https://github.com/ejescobedo/Lab-4/blob/master/lab4.py)

I, Edgar Escobedo, certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.