Nancy Hernandez

Diego Aguirre

Lab 1 Report

**Introduction**

For this lab I chose to code part B. The purpose of this lab was to implement a recursive method to generate all possible passwords using brute force. As well as to gain practice with recursion.

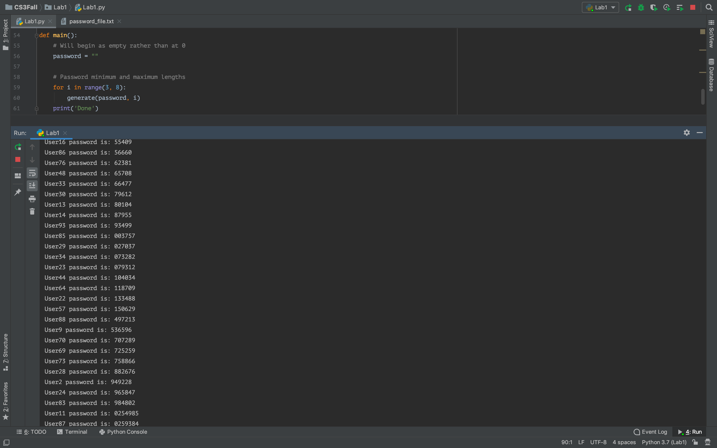
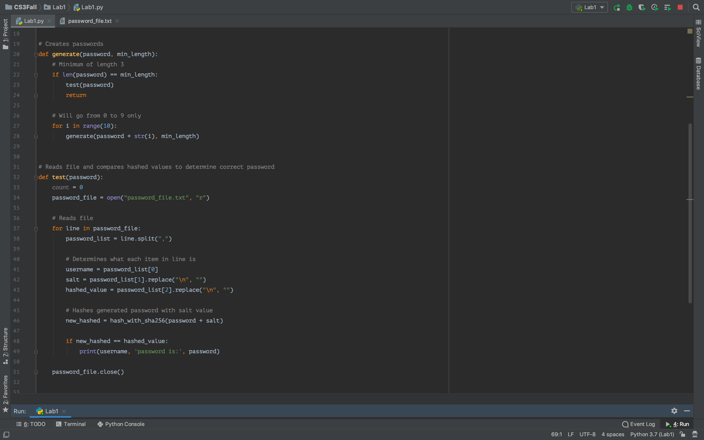
**Proposed Solution**

To begin, I thought it would be best to create the generate method. This method is in charge of actually creating the series of numbers that will be compared in the test method. The parameters for this particular method are the generated password and the minimum length. In order for the generated password to be sent to the test method, it has to be of at least a length of three. So, in this method I made an if statement that does just that. After the if statement comes the for loop that goes from numbers 0 to 9, and inside contains the recursive call.

Next, I created a method called test which is in charge of reading the file line by line and compare the hash values. With the generated password being the only parameter. In this case the hash values being compared will be the value already in the file and the new hashed value created by joining the generated password and the salt value. If when compared, are equal then the username and its corresponding password will be printed.

Finally, I have the main method. In this case the main has no parameters and has the generated password equal to nothing. Also, in this method there is a for loop that goes from 3 to 7 which contains the recursive call that increases the size. The reason for the password to be equal to nothing is because we can’t assume that the password isn’t empty.

**Results**

****

|  |  |
| --- | --- |
| **Number of Usernames** | **Running Time** |
| 100 | 4056 seconds |
| 50 | 1834 seconds |
| 25 | 1168 seconds |

**Test cases**:

UserTest1,abcdef,03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4  
UserTest2,ghijkl,fe2592b42a727e977f055947385b709cc82b16b9a87f88c6abf3900d65d0cdc3

I chose these test cases because I thought they would be good to experiment with.

**Conclusion**

From this lab I learned that this method of password hacking is not efficient as it takes a great amount of time. In addition, I also learned how use recursion to simplify certain tasks.

**Appendix**

Part B

import hashlib

def hash\_with\_sha256(str):

hash\_object = hashlib.sha256(str.encode(‘uf-8’))

hex\_dig = hash.object.hexdigest()

return hex\_dig

def main():

hed\_dig = hash\_with\_sha256(‘This is how you hash a string with sha256’)

print(hex\_dig)

main()

**Academic Dishonesty Statement**

I, Nancy Hernandez, was not involved in any copying from or providing information to another student, possessing unauthorized materials during a test, or falsifying data in laboratory reports. Neither did I participate in any type of collusion involving collaboration with another person to commit an academically dishonest act.