REPORT FOR DATA STRUCTURES LAB 1

In this lab I was asked to analyze 100 records containing usernames, salt values, and hashed passwords. My task was to create a python program that would read the txt file containing the data, analyze and compare each hashed password with the correct one. That is, finding the correct password. I was provided with the following restrictions:

* The method MUST be recursive
* Numbers 3 and 7 are not be hard-coded inside the method that generates the passwords; they should be parameters of the method
* You can have *at most* two nested loops inside this method (less than two is perfectly fine as well).

In order to revise the passwords, I first concatenated ‘s’ with the user’s salt value. By creating all possible permutations of a password ‘s’ with digits ranging from 0-9, I created a for loop that recursively calls “revise” which checks that the hash string is the correct length,

def revise(s, n):

    if(len(s) == n):

        # print(s)

        check(s)

        return

    for i in range(10):

        revise(s + str(i), n)

if so, it will call “check” which checks if the permutation passed matches any of the values in the file. “Check method works by split all values with each comma, and creating a new Hash by concatenating password permutation ‘s’ with the salt value., if they math, it will print “user password is (permutation ‘s’)” :

def check(s):

    file = open("password\_file.txt", "r")

    read = file.readlines()

    for line in read:

        arr = line.split(",")

        user = arr[0]

        salt\_value = arr[1]

        hashedpassfromarr = arr[2].replace('\n', '')

        newhash = concatinate(s, salt\_value)

        newhash = hash\_with\_sha256(newhash)

        if newhash == hashedpassfromarr:

            print(user + " password is: " + s)

        file.close()