CS5590/490-Python/DeepLearning

LAB ASSIGNMENT-1

By: Syed Jawad Hussain Shah

Student ID: 16117985

Class ID: 46

**Author:** Syed Jawad Hussain Shah

**Objective:**

The objective of this lab task is to get familiar with Python language and how to use this language to code for given tasks,

**Features:**

The features of this lab include validating password for its length and contents. Further, this also include finding middle word, longest word and reverse wording of a sentence. The third feature was to find the triplets in the list whose sum equals to 0. Finally, the last feature is related to find the common and not common elements between two lists.

**Configuration:**

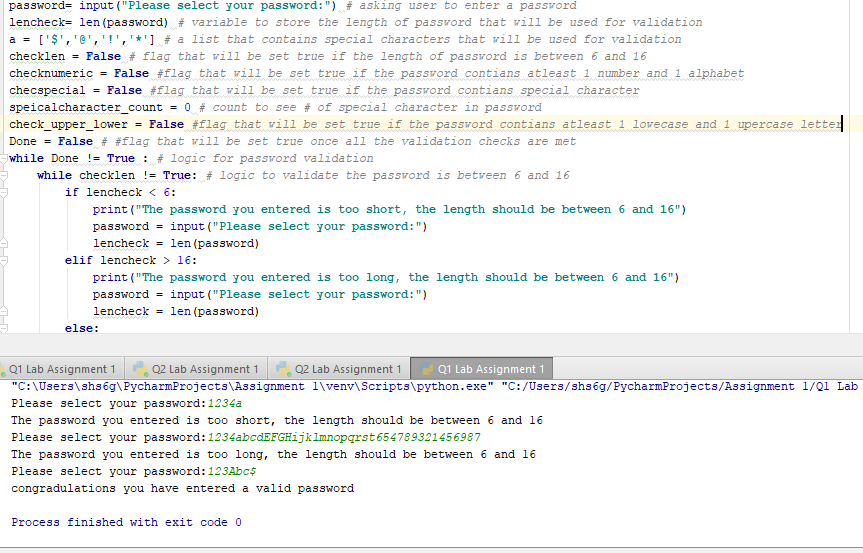
Python 3.6.4

IDE: JetBrains PyCharm community Edition 2017.3.3

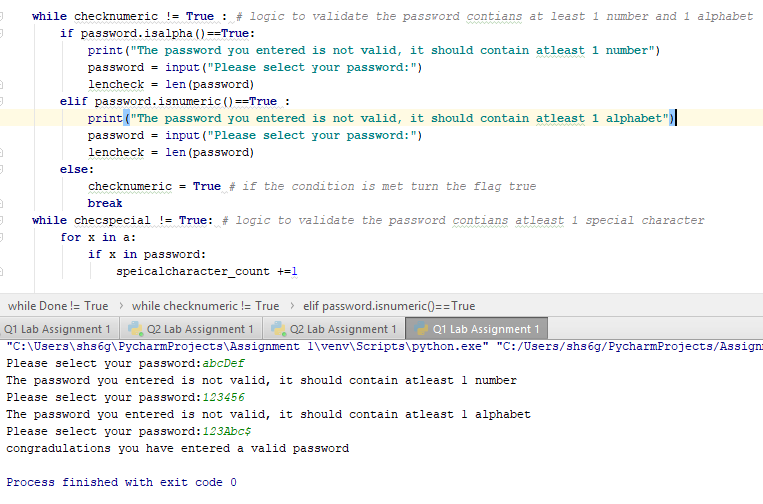
**Input/output (screenshots):**

Q1) Password validation:

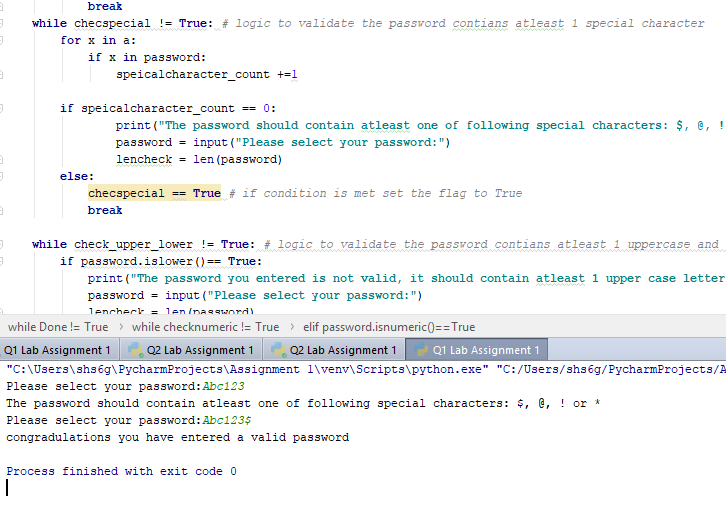
1. Length Validation:



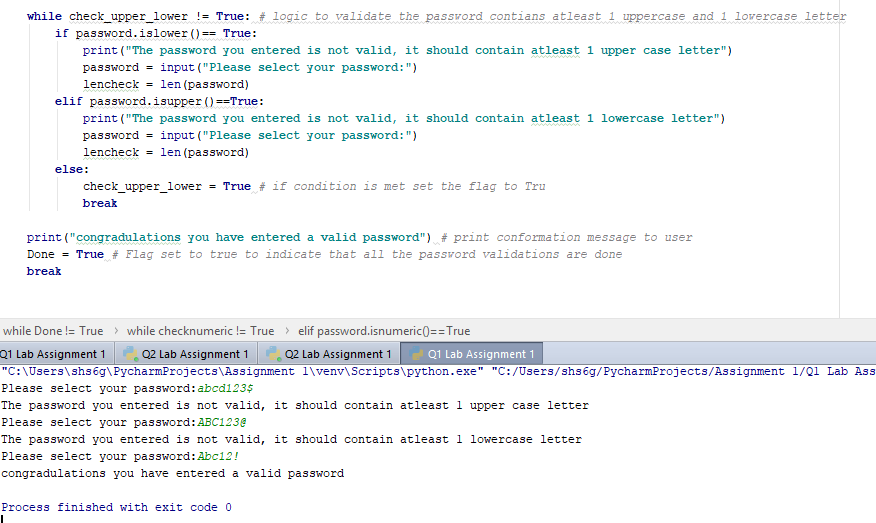
1. Number Validation:



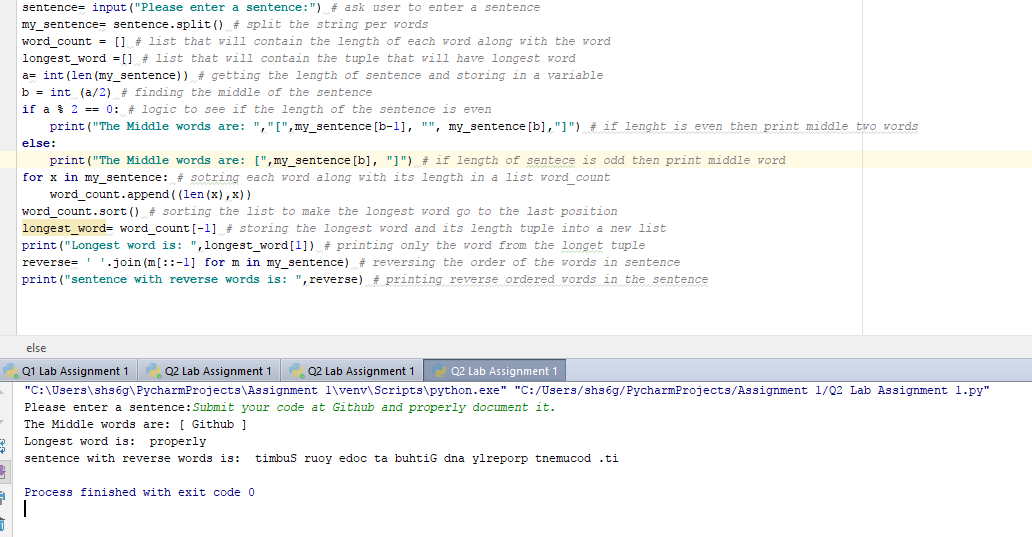
1. Special Character Validation:



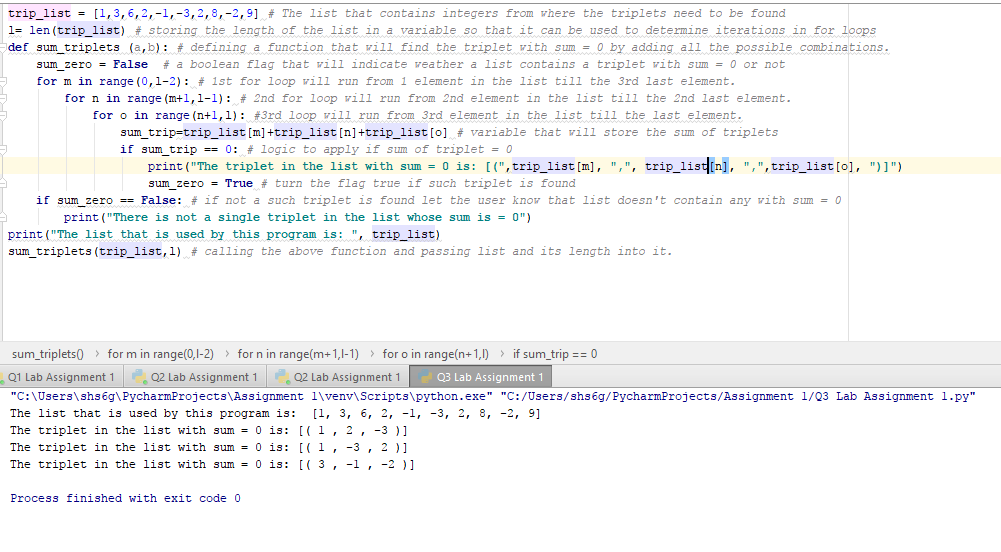
1. Uppercase or Lowercase Validation:



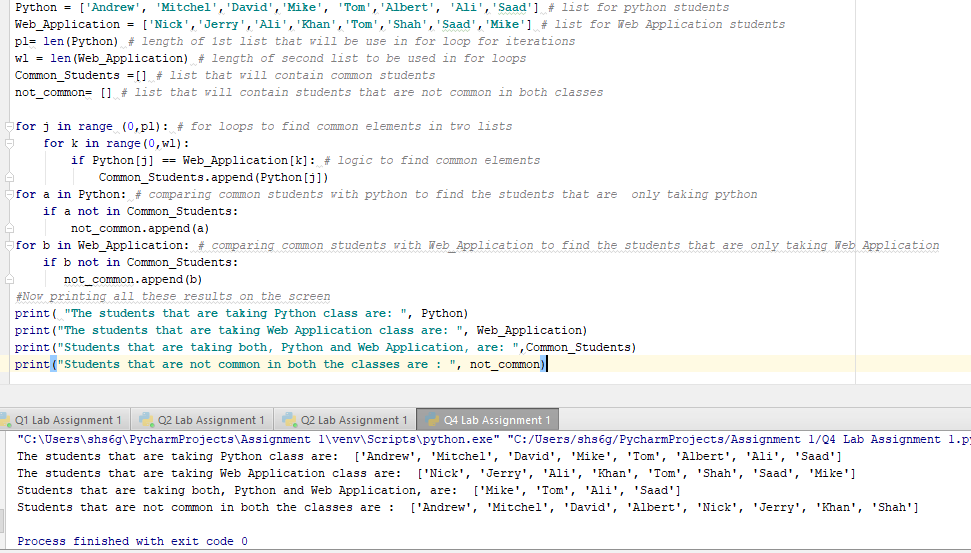
Q2) Middle word, longest word and reverse of the sentence screen shot:



Q3) Finding triplets in a list with sum equals 0:



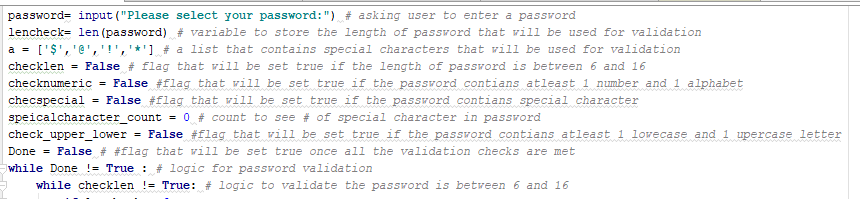
Q4) Finding common and uncommon students between two lists:



**Explanation of the implementation including code snippet**

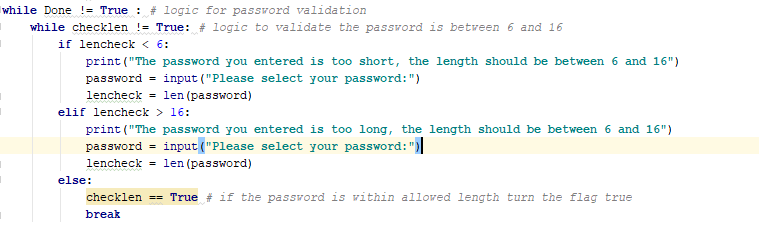
Q1) Password validation:

1. User input:



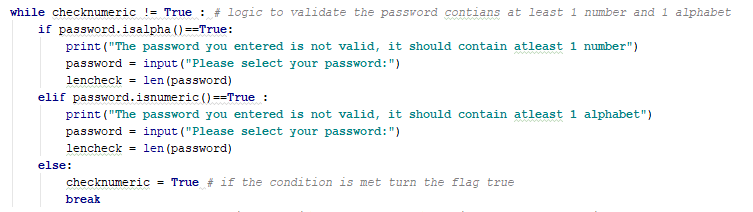
The program starts by asking user to input a password. After that it stores the password into a “password” variable and stores the length of the password in a “lencheck” variable. Then the validation flags (checklen, checknumeric, checkspecial, check\_upper\_lower, Done) are declared and set to false so that while loops can be run on the condition that these flags are not true. A list named ‘a’ stores the special character that will be validated against user input later in the program. Also, a variable called specialcharacter\_count is declared and set to 0. This variable is used later in program for special character validation. The first while loop executes, when it sees the Flag “Done” = False. This loop contains 4 nested loops to validate 4 requirements for the password.

1. Length Validation:



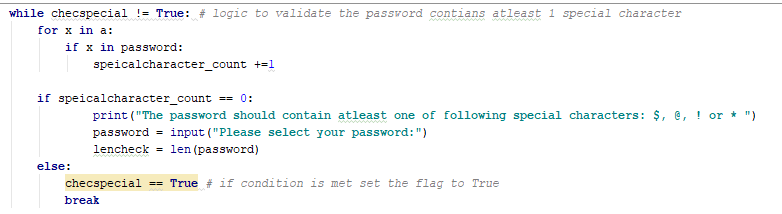
In this nested while loop there are two if statements. The first if statement checks if the length of the password is less than 6 and gives an error to the user and ask user to re-enter password between length of 6 and 16. The second if statement check if the length of the password is greater than 16 and gives an error to the user and ask user to re-enter password between length of 6 and 16. Once these checks are validated, the program goes into else statemen where the “checklen” Flag is switched to True and the nested loop is terminated.

1. Number Validation:



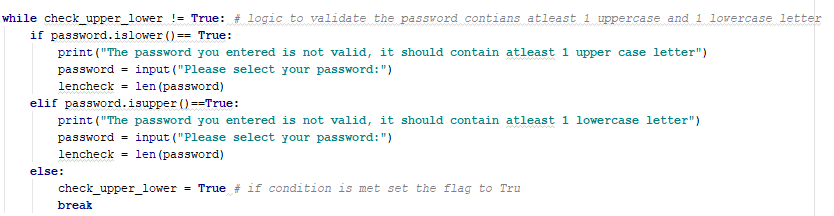
In this nested while loop there are two if statements. The first if statement checks if the password entered by the user contains all the alphabets if so then it gives an error to the user and ask user to re-enter password that contains at least 1 number. The second if statement checks if the password entered by the user contains all the numbers if so then it gives an error to the user and ask user to re-enter password that contains at least 1 alphabet. Once these checks are validated, the program goes into else statemen where the “checknumeric” Flag is switched to True and the nested loop is terminated.

1. Special Character Validation:



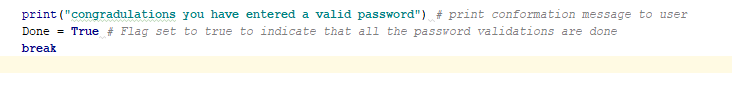
In this nested while loop there is a nested for loop. With in the for loop, programs count the number of special characters that are from list “a” and are present in the password. This count is stored into “specialcharacter\_count” variable. At the end of the for loop there is an if statement that checks the count of special character in the password. If this count is 0 then the program gives an error to the user and ask user to re-enter password that contains at least 1special character from list “a”. Once this check is validated, the program goes into else statemen where the “checkspecial” Flag is switched to True and the nested loop is terminated.

1. Upper-case and lower-case validation:



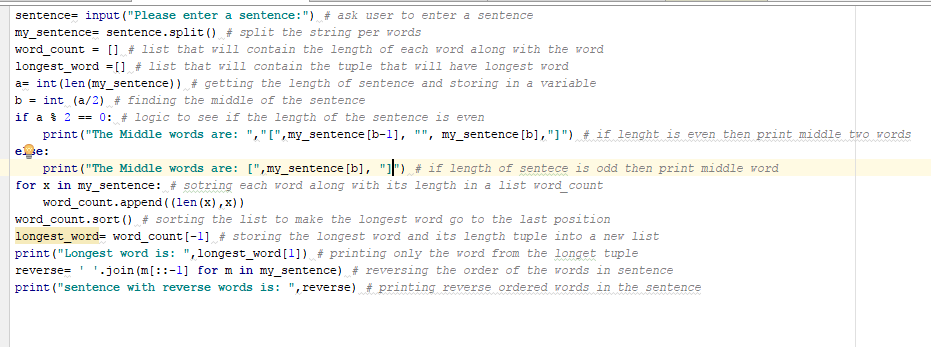
In this nested while loop there are two if statements. The first if statement checks if the password entered by the user contains all the lower-case letters if so then it gives an error to the user and ask user to re-enter password that contains at least 1 upper-case letter. The second if statement checks if the password entered by the user contains all the upper-case letter if so then it gives an error to the user and ask user to re-enter password that contains at least 1 lower-case letter. Once these checks are validated, the program goes into else statemen where the “check\_upper\_lower” Flag is switched to True and the nested loop is terminated.

1. The final conformation:



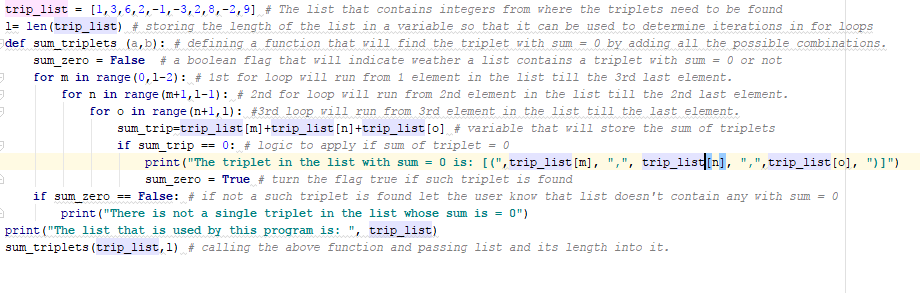
In this part, once all the validations are met, a conformation message is presented to the user. The “Done” Flag is switched to True and the very first while loop as well as program is terminated.

Q2) Finding the middle and longest word and reverse word order of the sentence:



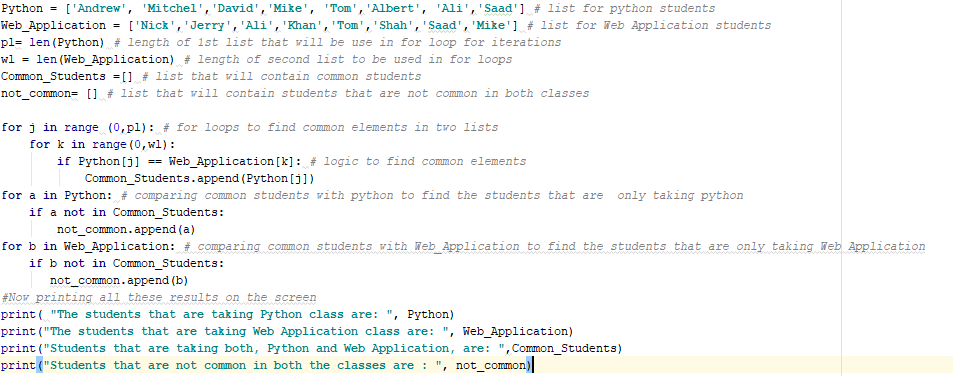
The program starts by asking the user to enter a sentence and stores the input in a variable called “sentence”. Then this string is break up into words by using .split function and stored into “my\_sentence” variable. After that an empty list is declared with the name “word\_count”. This list will be used later by the program to store words along with their respective length. Another empty list with the name “longest\_word” is created that will be use later by the program to store the tuple that contains the longest word along with its respective length. A variable “a” is declared to store the length of my\_sentence. A variable “b” is declared to find the middle index of the string. The if statement checks if the length of sentence is even, if so it will print the middle two words. The else statement checks if the length of the sentence is odd, if so it will print the middle word. The for loop appends the word\_count list with words and their respective length. After this .sort function is used to sort this list so that longest word goes to the last of the list. Longest\_word list is set to the last tuple in word\_count and the longest word is printed. Finally, the reverse of the words in sentence is found by using. join[::-1] function and is printed.

Q3) Finding triplets in the list that sums to 0:



In this question a simple technique of 3 for loops is used to find the triplets in the list with sum equals to zero. The first for loop runs from first index to the 3rd last index in the list, the second for loop runs from 2nd index in the list to second last index in the list, the last from loop runs from 3rd index in the list to last index in the list, these three loops adds all the possible combinations of the triplets in the list and finds the ones whose sum is equal to zero. Once such triplets are found they are printed on the screen. All this logic is stored into a function called “sum\_triplets”. This function takes two input a list and its length. The length of the list is used to set the upper limit on the for loops.

Q4) Finding the common and uncommon students in two classes:



In this program two list, Python, and Web Application, stores the name of the students who are taking these courses. pl stores the length of python list, and wl stores the length of web application list. Common\_students is the list that will contain the students that are common in both classes. Not\_common is the list that will contain students that are not common in both classes. For loop is used to find the common students in both classes and once such student is found common\_student is appended with the student name. once all the common students are found and stored into a common\_student list, this list is compared with name of students in python and web application to find students that are in those courses but not in common list. Once such students are found not\_common list is appended with their name. At the end the results are printed.

**Deployment:**

1. Save the folder in your local machine.
2. Install Python 3.6.4 and PyCharm IDE on your machine.
3. Run PyCharm, click on file->open->files location of the saved folder.
4. Select the desired code file with .py extension to execute.
5. Right click on the code screen and then click on run "filename".
6. Give the input and validate the output.

**limitation**

1. The password program only accepts passwords with special character, length between 6 and 16, with containing at least 1 number and 1 alphabet and at least 1 upper case and 1 lowercase letter. However, once these individual validations are passes, they are not revalidated in case user enters a new password as result of failure down stream of the previous validation check.
2. Paragraphs and multipage documents are not recommended for word find program in Q2.
3. The complexity is n3, which will impact the performance of the program in case of larger input.
4. Program 4 assumes that the user data will be provided and doesn’t generates its own data.

**References**

<https://stackoverflow.com/questions/18791882/python-how-to-make-program-go-back-to-the-top-of-the-code-instead-of-closing>

<https://stackoverflow.com/questions/41228115/how-to-extract-the-first-and-final-words-from-a-string>

<https://www.saltycrane.com/blog/2009/04/how-reverse-words-sentence-using-python-and-c/>

<https://stackoverflow.com/questions/8458434/reverse-each-word-in-a-string>

<https://www.geeksforgeeks.org/find-triplets-array-whose-sum-equal-zero/>