CSEE 5590 0001
Special Topics SPRING 2018

PHYTHON LAB ASSIGNMENT -1

Submitted On 2/2/2017

Name: Satya Sai Deepthi Katta

Class ID: 21

UMKC

Department of Computer Science and Electrical Engineering

DOCUMENT CONTENTS

AUTHOR

This document is a part of Lab Assignment #1 submitted by SATYA SAI DEEPTHI KATTA (ID: 21), a graduate student majoring in computer science at University of Missouri Kansas City. The lab assignment carried out under Python and Deep Learning course(CS5590) taught by Dr. Yungyug Lee along with Rashmi, Saira and Vijaya Yeruva.

OBJECTIVE

The main aim of the lab work is to create an exposure to some basic concepts in Python language like:

- Lists
- Conditional Statements
- Loops …

The assignment is divided into four tasks which focuses to make one familiar with basic python concepts listed above.

- To validate password with specified requirements.
- To print middle words of given sentence along with printing longest word of sentence and to reverse given sentence.
- To print triple which sum to zero in given list
- To print list of students who have common classes in Python and Web and print list of students who are not common in both classes.

FEATURES

The features involved in each task are documented below.

Task 1:

Validate Password

In any web application the user needs to login using a valid password. For the UMKC Web application, the following are requirements for valid password.

a) password length must be between 6 and 16 characters long.

b) it should have at least one numeric value.

c) password should contain a special character in the list of $, @, !, *

d) password should contain at least one upper and lower case letter.

The task to python function using loops to set a valid password for user login.

Task 2:

Printing form Sentence

The task to write a python function which gives takes input sentence from user and prints

- middle words of given sentence
- longest word among all the listed words in sentence
- reversed sentence with all characters from left to right.

Task 3:

Triplet sum to zero

This task finds triplets whose sum will be zero from the list of numbers given by the user.

Task 4:

Printing common and uncommon lists

The scenario comprises of two lists of students attending two classes namely Python and Web. The task to find the common students attending both the classes and print them in a list. Along with this, a separate list needs to printed displaying the students who don’t attend both the classes.

CONFIGURATION

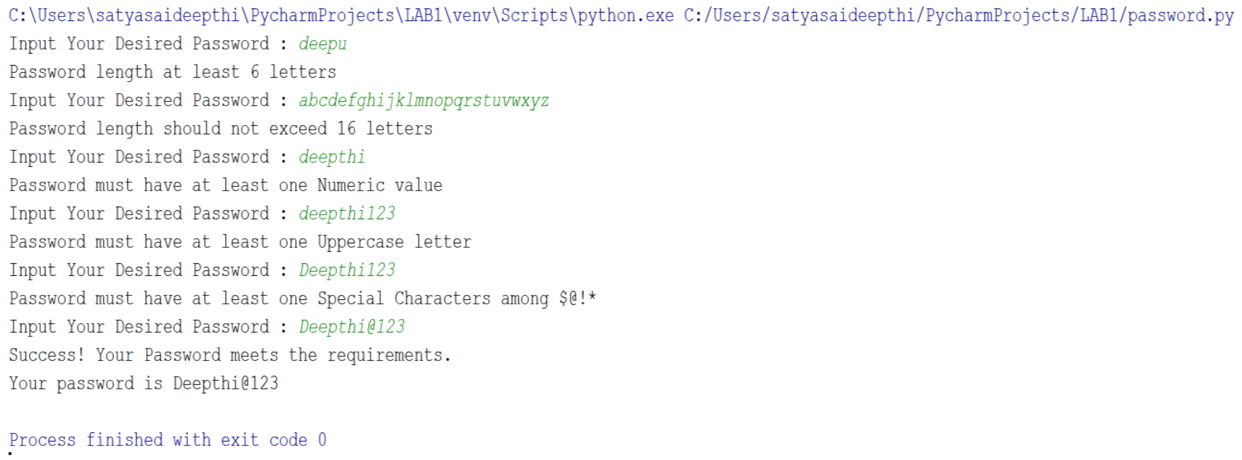
For executing the tasks given in the lab assignment, advanced version of **Python 3.6.4** is used and the code is built in **PYCHARM** Software.

INPUT/OUTPUT SCREENSHOTS

Task 1:

For UMKC web application, the password should meet some requirements to be accepted. A desired password is entered by the user and code iterates itself until the desired password meets the requirements of the valid password and prints it if it is a success.

The screenshot shows the input being entered by the user in green and the modifications are given. The output is shown with a Success message and printing the valid password which meets the given details.

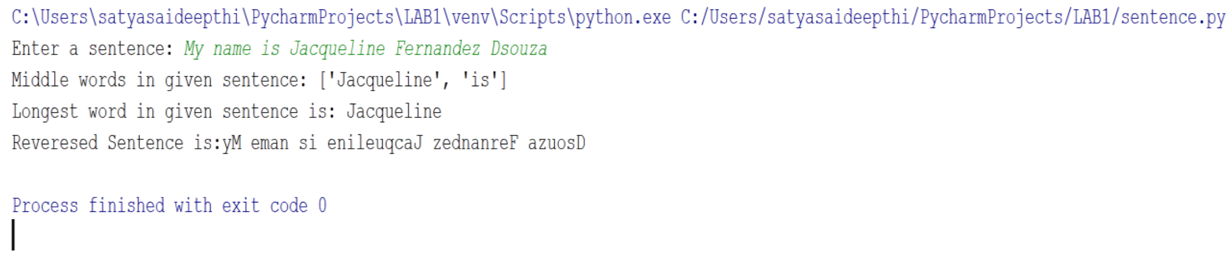


```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/password.py
Input Your Desired Password : deepu
Password length at least 6 letters
Input Your Desired Password : abcdefghijklmnopqrstuvwxyz
Password length should not exceed 16 letters
Input Your Desired Password : deepthi
Password must have at least one Numeric value
Input Your Desired Password : deepthi123
Password must have at least one Uppercase letter
Input Your Desired Password : Deepthi123
Password must have at least one Special Characters among $@!*
Input Your Desired Password : Deepthi@123
Success! Your Password meets the requirements.
Your password is Deepthi@123

Process finished with exit code 0
```

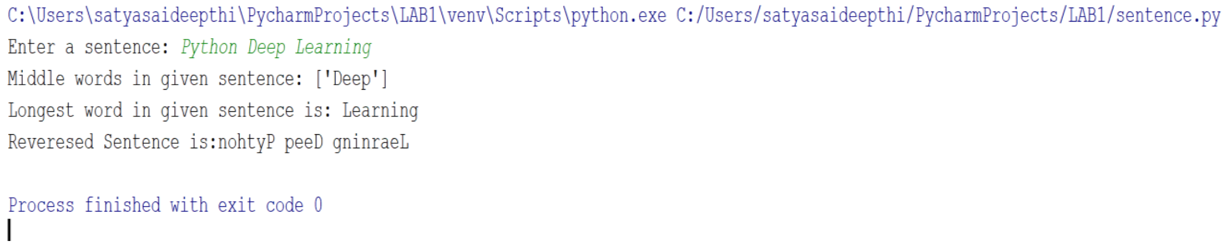
Task 2:

Here, a sentence is taken from the user as input. Three different outputs are printed by running a single function.

- Firstly, to print the middle word of the sentence if it has odd number of words and to print to middle words if it has even number of words in a sentence.
- The second output is to print the longest word among all the words in the given sentence.
- Finally, to print a reversed sentence is printed reversing all the words of the given sentence while keeping the sentence structure as given.



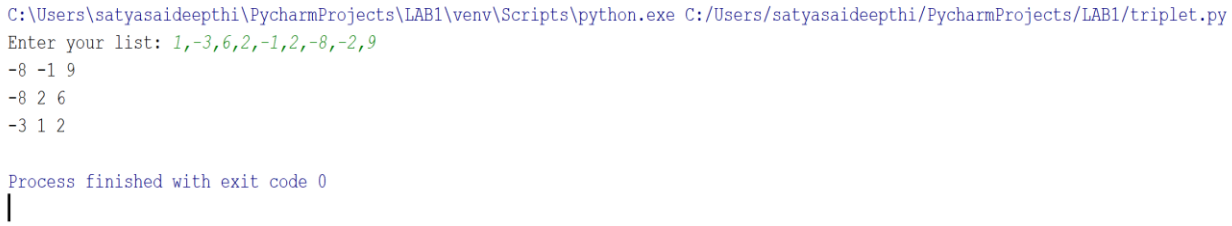
```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/sentence.py
Enter a sentence: My name is Jacqueline Fernandez Dsouza
Middle words in given sentence: ['Jacqueline', 'is']
Longest word in given sentence is: Jacqueline
Reveresed Sentence is:yM eman si enileuqcaJ zednanreF azuosD

Process finished with exit code 0
```



```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/sentence.py
Enter a sentence: Python Deep Learning
Middle words in given sentence: ['Deep']
Longest word in given sentence is: Learning
Reveresed Sentence is:nohtyP peeD gninraeL

Process finished with exit code 0
```

Task 3:

This task takes input a list of numbers separated by “,”. List of numbers who sum gives a zero are printed as output.

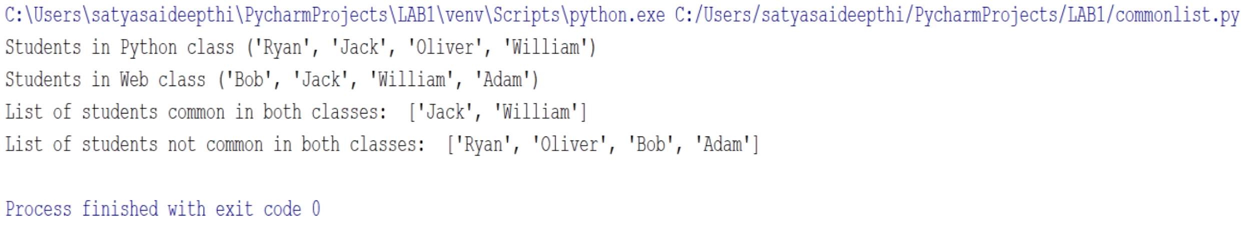


```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/triplet.py
Enter your list: 1,-3,6,2,-1,2,-8,-2,9
-8 -1 9
-8 2 6
-3 1 2

Process finished with exit code 0
```

Task 4:

Python and Web classes student list are given as an input for this task. The output comprises of two lists.

- One of them printing the list of students who have common classes in both the courses mentioned.
- While the other list is to print remaining students who does not have any common between the courses.

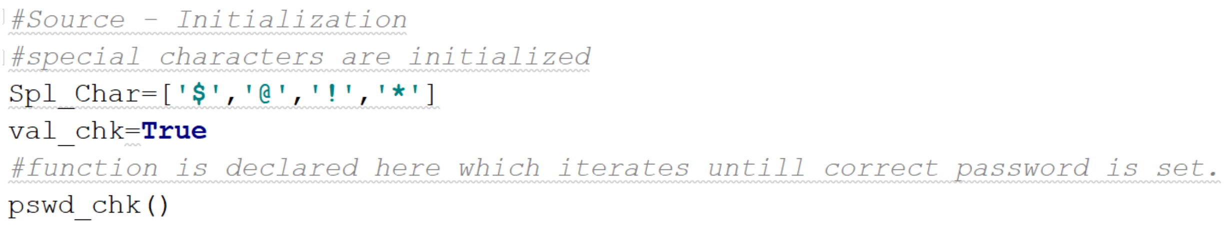


```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/commonlist.py
Students in Python class ('Ryan', 'Jack', 'Oliver', 'William')
Students in Web class ('Bob', 'Jack', 'William', 'Adam')
List of students common in both classes:  ['Jack', 'William']
List of students not common in both classes:  ['Ryan', 'Oliver', 'Bob', 'Adam']

Process finished with exit code 0
```

IMPLEMENTATION & CODE SNIPPET

Task 1:

Python code to validate a password starts with initializing special characters given in the requirements i.e., [$, @, !, *] . Then a Boolean return value is given value True. Now a function is written to set a valid password for UMKC Web application.

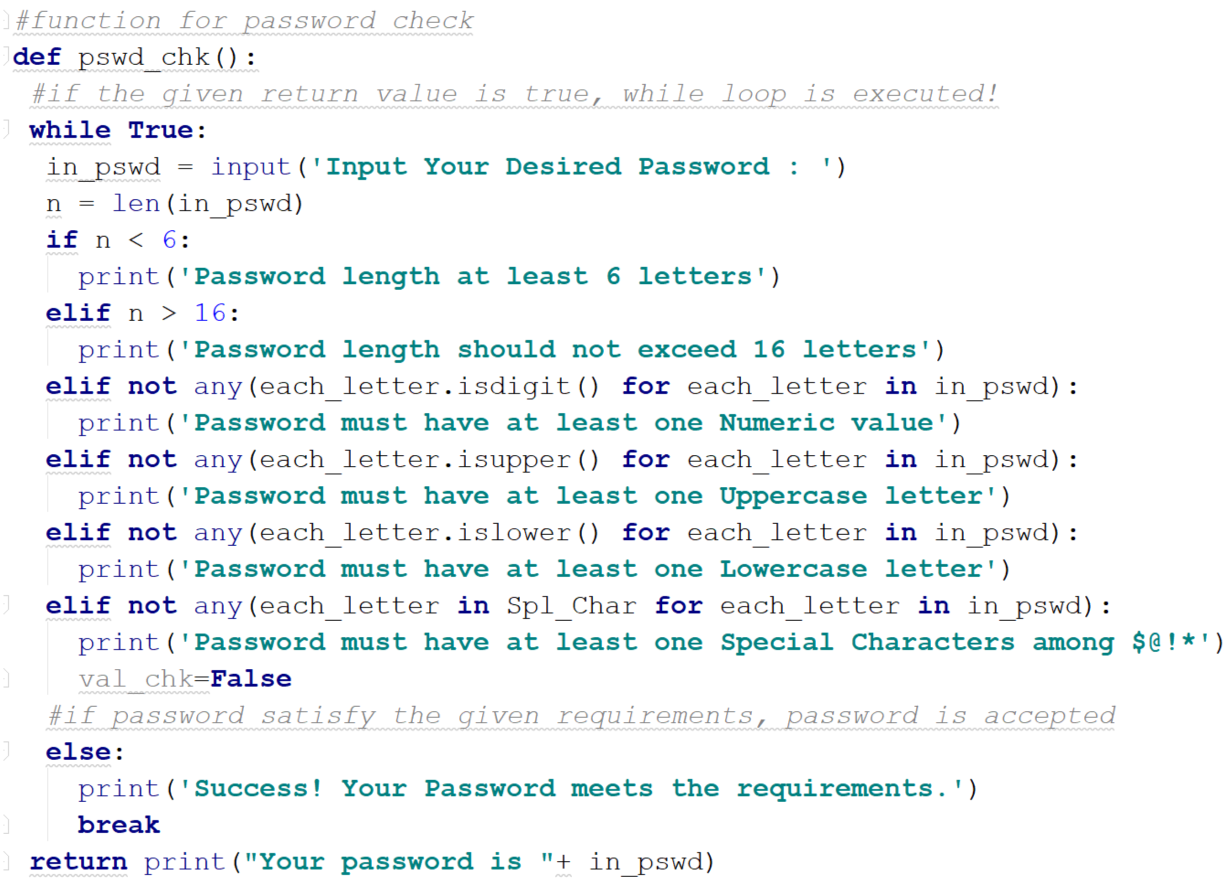


```
#Source - Initialization
#special characters are initialized
Spl_Char=['$','@','!','*']
val_chk=True
#function is declared here which iterates untill correct password is set.
pswd_chk()
```

The python function **pswd_chk()** validates the user desired input password by iterating a while loop which checks for return value. If return value is true, then the loop is executed asking the user for a desired input.

Now, if else statements are used to check for the requirements namely, at least one uppercase, numeric value, and length of password should be in range of 6-16, mostly importantly containing one of the special characters stated.

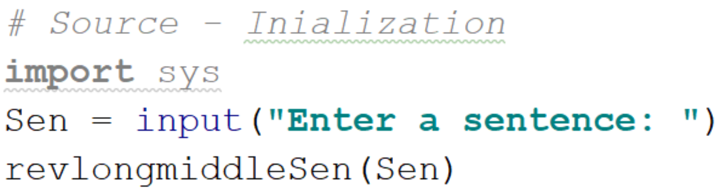
If all the conditions are satisfied, a success message is printed by breaking the while loop. The function returns a print statement with the valid password best suited for the application.



```
#function for password check
def pswd_chk():
    #if the given return value is true, while loop is executed!
    while True:
        in_pswd = input('Input Your Desired Password : ')
        n = len(in_pswd)
        if n < 6:
            print('Password length at least 6 letters')
        elif n > 16:
            print('Password length should not exceed 16 letters')
        elif not any(each_letter.isdigit() for each_letter in in_pswd):
            print('Password must have at least one Numeric value')
        elif not any(each_letter.isupper() for each_letter in in_pswd):
            print('Password must have at least one Uppercase letter')
        elif not any(each_letter.islower() for each_letter in in_pswd):
            print('Password must have at least one Lowercase letter')
        elif not any(each_letter in Spl_Char for each_letter in in_pswd):
            print('Password must have at least one Special Characters among $@!*')
            val_chk=False
        #if password satisfy the given requirements, password is accepted
        else:
            print('Success! Your Password meets the requirements.')
            break
    return print("Your password is "+ in_pswd)
```

Task 2:

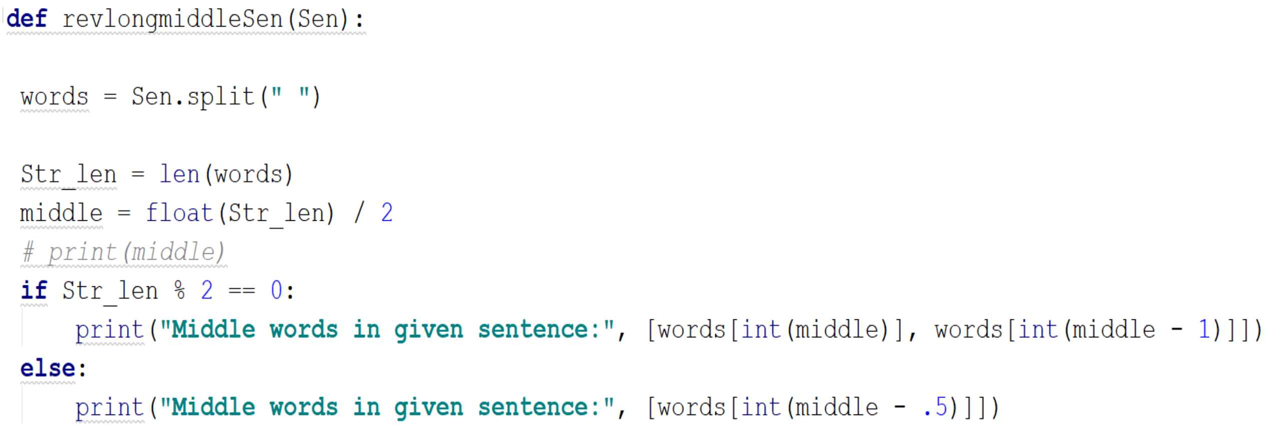
This task import system specification library as the input is taken from the user using built-in function input(). The python function to print the listed output of the task are written in the function defining **revlongmiddleSen()** with a input parameter of sentence given by the user.



```
# Source - Inialization
import sys
Sen = input("Enter a sentence: ")
revlongmiddleSen(Sen)
```

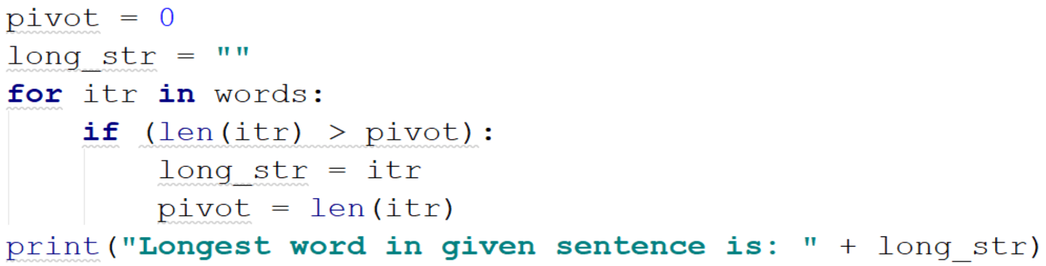
In revlongmiddleSen(), firstly words in a sentence are separated into list form by splitting them using space between them. Total word count of the sentence is stored in str_len.

To find middle words, a middle parameter is considered which calculates float value by diving number of words by 2. If condition statements is used to print the middle words of the sentence if the words are odd, while else condition is executed if the words are even.



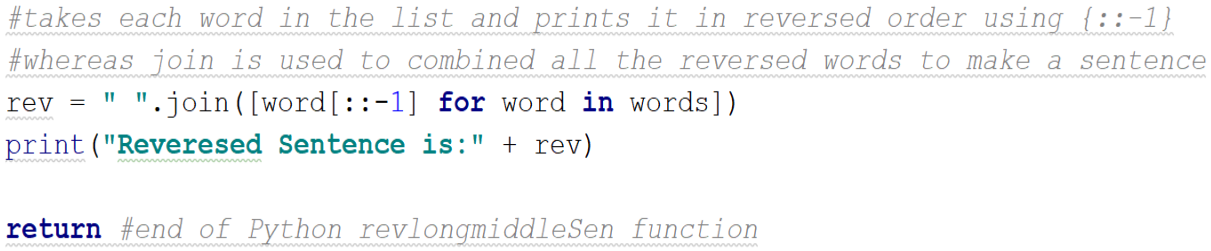
```
def revlongmiddleSen(Sen):

    words = Sen.split(" ")

    Str_len = len(words)
    middle = float(Str_len) / 2
    # print(middle)
    if Str_len % 2 == 0:
        print("Middle words in given sentence:", [words[int(middle)], words[int(middle - 1)]])
    else:
        print("Middle words in given sentence:", [words[int(middle - .5)]])
```

For printing the longest word of the sentence, a for is iterated taking the length of each word and storing them in two different parameters. Here, pivot is used to store the length of the word being iterated. While long_str stores the value of the longest word value and keep changing it if it encounters longer word length than one being stored.



```
pivot = 0
long_str = ""
for itr in words:
    if (len(itr) > pivot):
        long_str = itr
        pivot = len(itr)
print("Longest word in given sentence is: " + long_str)
```

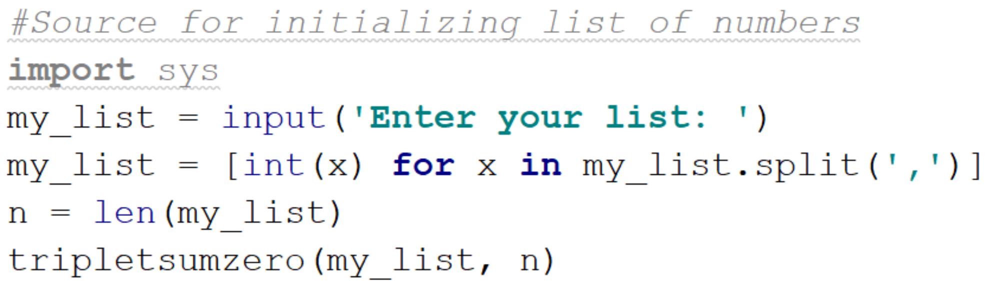
To reverse a word is done by using word[::-1], prints the words right to left. In the code .join() is used for combing all the words and making them into a sentence leaving space between the reversed words.



```
#takes each word in the list and prints it in reversed order using {::-1}
#whereas join is used to combined all the reversed words to make a sentence
rev = " ".join([word[::-1] for word in words])
print("Reveresed Sentence is:" + rev)

return #end of Python revlongmiddleSen function
```

Task 3:

The user defined list of numbers is given using a separator (,) as input for the task. A parameter ‘n’ stores the value of total number of words given. A python function **tripletsumzero()** inputting parameters of the user defined list along with number count.



```
#Source for initializing list of numbers
import sys
my_list = input('Enter your list: ')
my_list = [int(x) for x in my_list.split(',')]
n = len(my_list)
tripletsumzero(my_list, n)
```

In tripletsumzero() function, firstly found variable is initialized to False Boolean value.

Then the list given by user is sorted into an array.

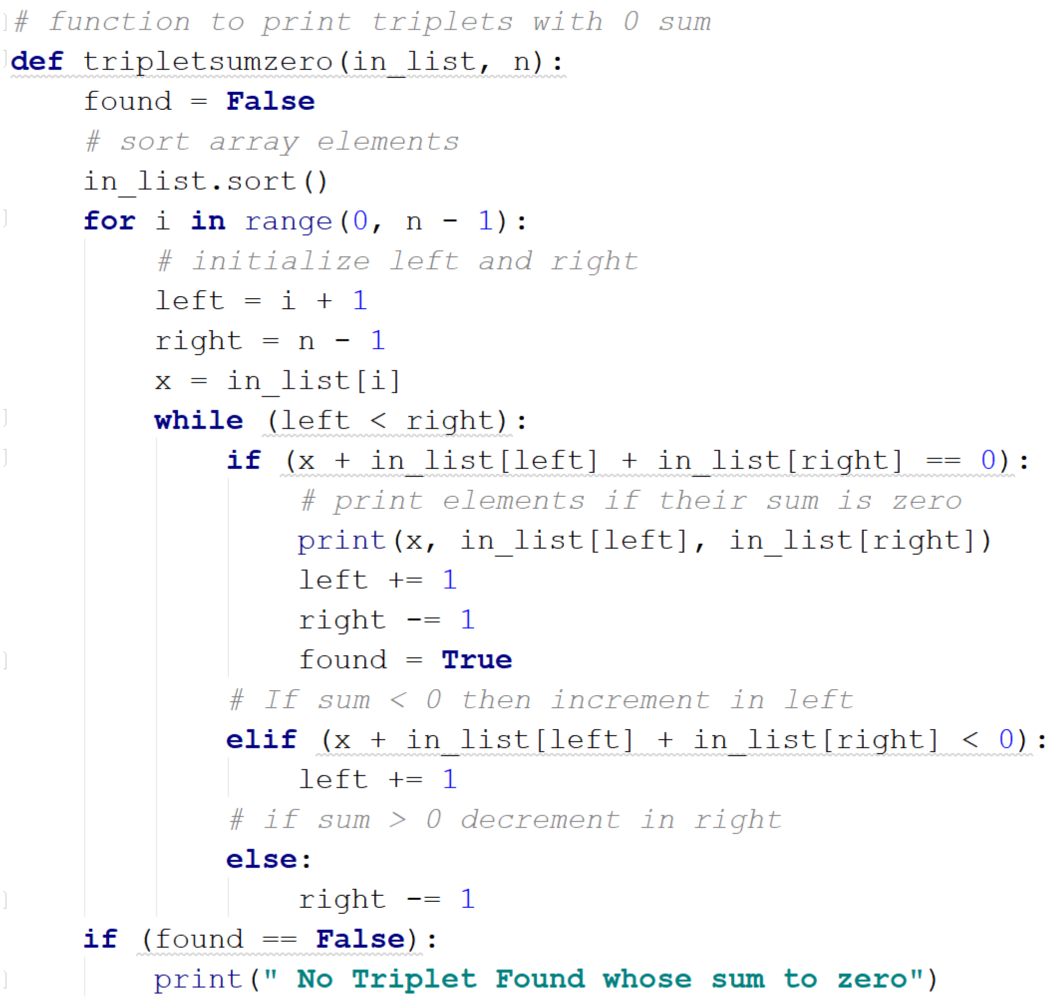
Now a loop is run from 0 to n-2 elements, initializing the index variables representing left and right.

A while loop is considered to run a decision is left is less than right.

Inside a loop, sum of the elements is calculated which are present in left, right and ‘i’th position.

- If the sum is zero then counter increments left list and decrements right counter and found value is turned to True, printing the triplets.
- If the sum is less than zero then left value counter is incremented and the right value remains the same.
- If the sum is greater than zero then the right counter is decremented while the left value remains unchanged.

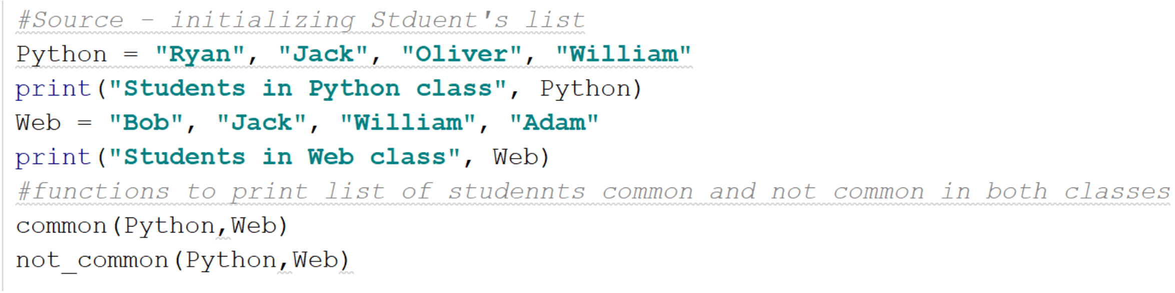
If there is no triplet found in the list given, then the ‘found’ value remains constant throughout the loop i.e. False, so if loop at the end of function is iterated printing ‘No Triplet Found whose Sum is Zero’



```
# function to print triplets with 0 sum
def tripletsumzero(in_list, n):
    found = False
    # sort array elements
    in_list.sort()
    for i in range(0, n - 1):
        # initialize left and right
        left = i + 1
        right = n - 1
        x = in_list[i]
        while (left < right):
            if (x + in_list[left] + in_list[right] == 0):
                # print elements if their sum is zero
                print(x, in_list[left], in_list[right])
                left += 1
                right -= 1
                found = True
            # If sum < 0 then increment in left
            elif (x + in_list[left] + in_list[right] < 0):
                left += 1
            # if sum > 0 decrement in right
            else:
                right -= 1
    if (found == False):
        print(" No Triplet Found whose sum to zero")
```

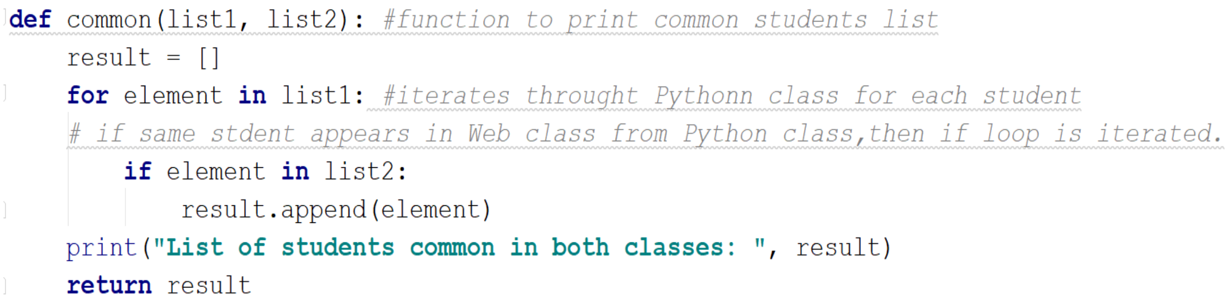
Task 4:

List of students enrolled in the courses Python and Wed Application are given as input to the task. A function **common()** with input parameters of Python and Web students list is developed to find out the common students enrolled in both the classes. Whereas **not_common()** function is defined with same input parameters to find and print the list of students who don’t have classes in common.



```
#Source - initializing Stduent's list
Python = "Ryan", "Jack", "Oliver", "William"
print("Students in Python class", Python)
Web = "Bob", "Jack", "William", "Adam"
print("Students in Web class", Web)
#functions to print list of studennts common and not common in both classes
common(Python,Web)
not_common(Python,Web)
```

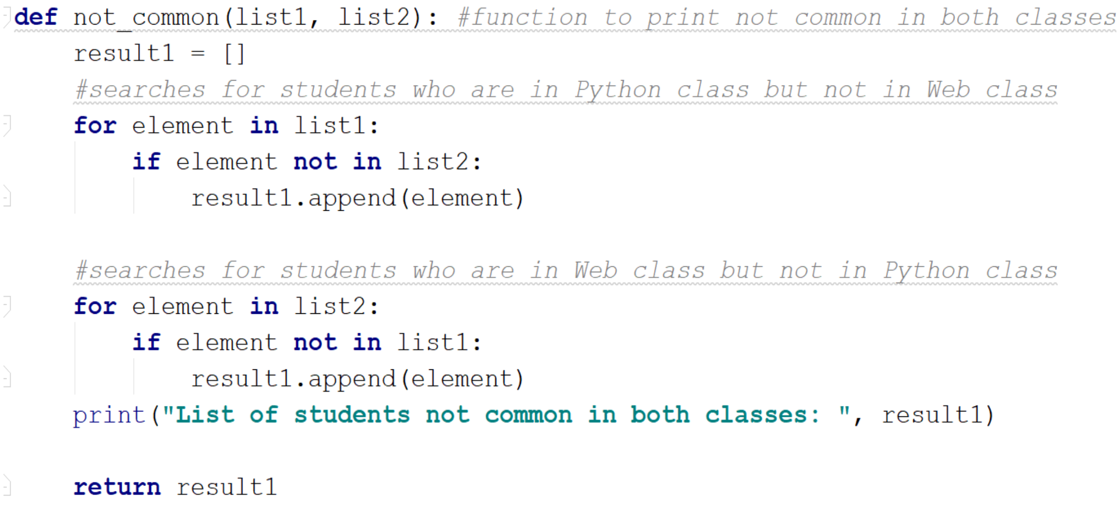
Two loops are used to build common() function and the names of all common students is stored in a list name ‘result’. Every element in Python list is compared with every element of Web list, if a match is found then the name is appended to result list and printed at the end.



```
def common(list1, list2): #function to print common students list
    result = []
    for element in list1: #iterates throught Pythonn class for each student
    # if same stdent appears in Web class from Python class,then if loop is iterated.
        if element in list2:
            result.append(element)
    print("List of students common in both classes: ", result)
    return result
```

Not_common() is designed using the same concept of using two loops and storing the result as a list. But instead of comparing and finding same elements of list, non-common elements are appended into the result1 list.

In the first loop, Python list is iterated against Web then the non-common student’s names from python are stored in result1. In the second loop, Web list is iterated against Python and the students names from Web list are appended to the list of students already containing in result1.



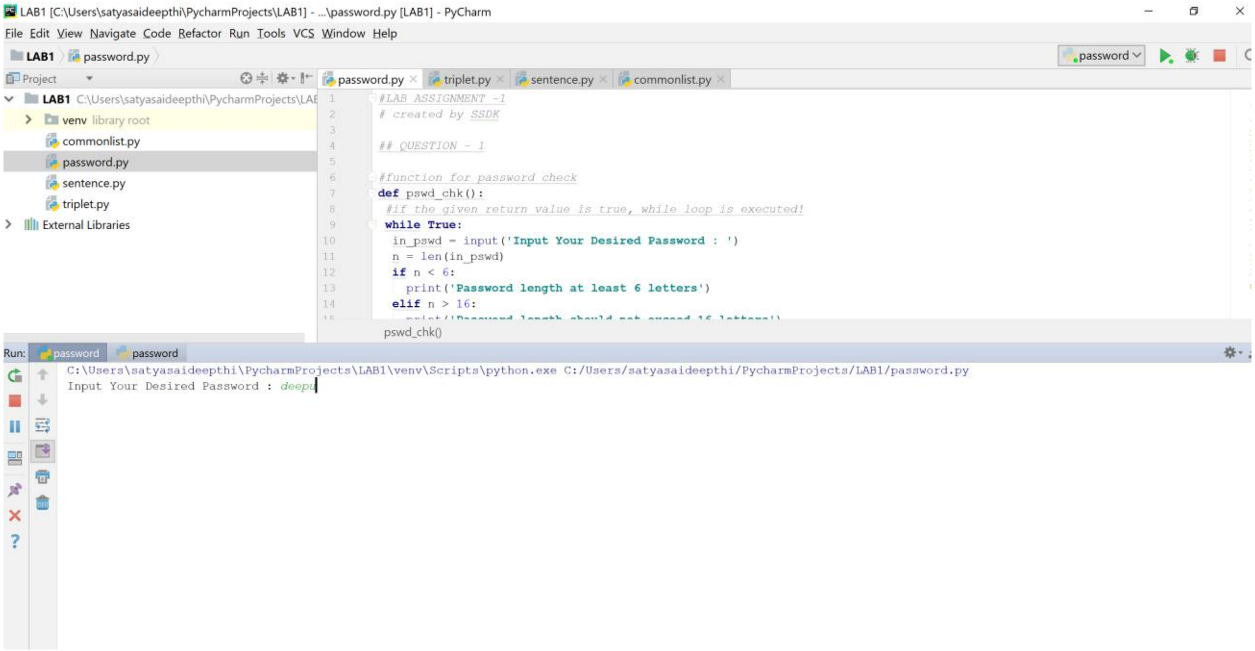
```
def not_common(list1, list2): #function to print not common in both classes
    result1 = []
    #searches for students who are in Python class but not in Web class
    for element in list1:
        if element not in list2:
            result1.append(element)

    #searches for students who are in Web class but not in Python class
    for element in list2:
        if element not in list1:
            result1.append(element)
    print("List of students not common in both classes: ", result1)

    return result1
```

DEPLOYMENT

For this assignment, all the code is deployed on PYCHARM software saving all the program files in .py format. The console window is used for giving the inputs and for getting the outputs from the code.

Task 1 Deployment:

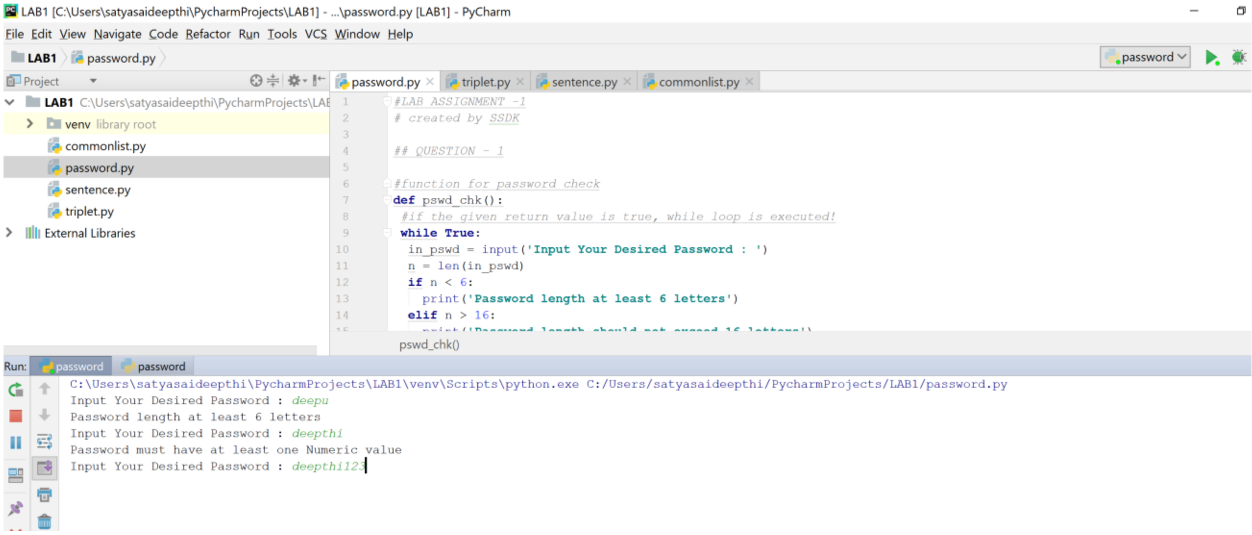
Step1: Insert user desired password


LAB1 [C:\Users\satyasaideepthi\PycharmProjects\LAB1] - ...\password.py [LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LAB1 password.py
password
Project
LAB1 C:\Users\satyasaideepthi\PycharmProjects\LAB
venv library root
commonlist.py
password.py
sentence.py
triplet.py
External Libraries
password.py triplet.py sentence.py commonlist.py
pswd_chk()
Run: password password

```
#LAB ASSIGNMENT -1
# created by SSDK

## QUESTION - 1

#function for password check
def pswd_chk():
 #if the given return value is true, while loop is executed!
 while True:
  in_pswd = input('Input Your Desired Password : ')
  n = len(in_pswd)
  if n < 6:
   print('Password length at least 6 letters')
  elif n > 16:
   print('Password length should not exceed 16 letters')
```

```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/password.py
Input Your Desired Password : deepu
```

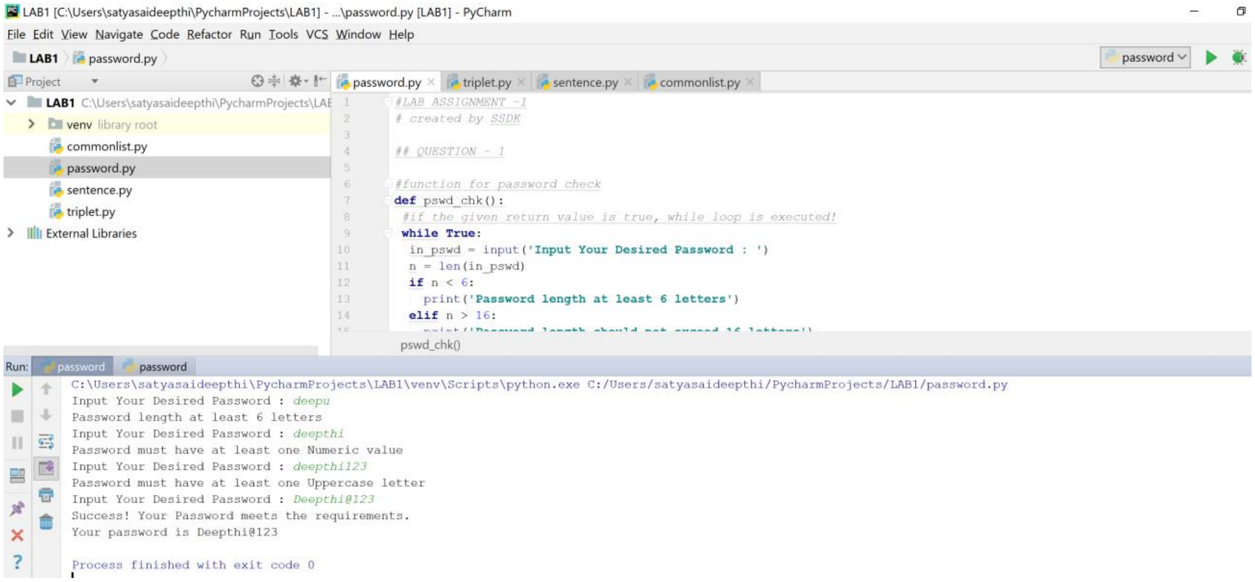
Step 2: Modify the password based on the changes specified


LAB1 [C:\Users\satyasaideepthi\PycharmProjects\LAB1] - ...\password.py [LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LAB1 password.py
password
Project
LAB1 C:\Users\satyasaideepthi\PycharmProjects\LAB
venv library root
commonlist.py
password.py
sentence.py
triplet.py
External Libraries
password.py triplet.py sentence.py commonlist.py
pswd_chk()
Run: password password

```
#LAB ASSIGNMENT -1
# created by SSDK

## QUESTION - 1

#function for password check
def pswd_chk():
 #if the given return value is true, while loop is executed!
 while True:
  in_pswd = input('Input Your Desired Password : ')
  n = len(in_pswd)
  if n < 6:
   print('Password length at least 6 letters')
  elif n > 16:
   print('Password length should not exceed 16 letters')
```

```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/password.py
Input Your Desired Password : deepu
Password length at least 6 letters
Input Your Desired Password : deepthi
Password must have at least one Numeric value
Input Your Desired Password : deepthi123
```

Step 3: Changes need to be made until a success note appears on console window.

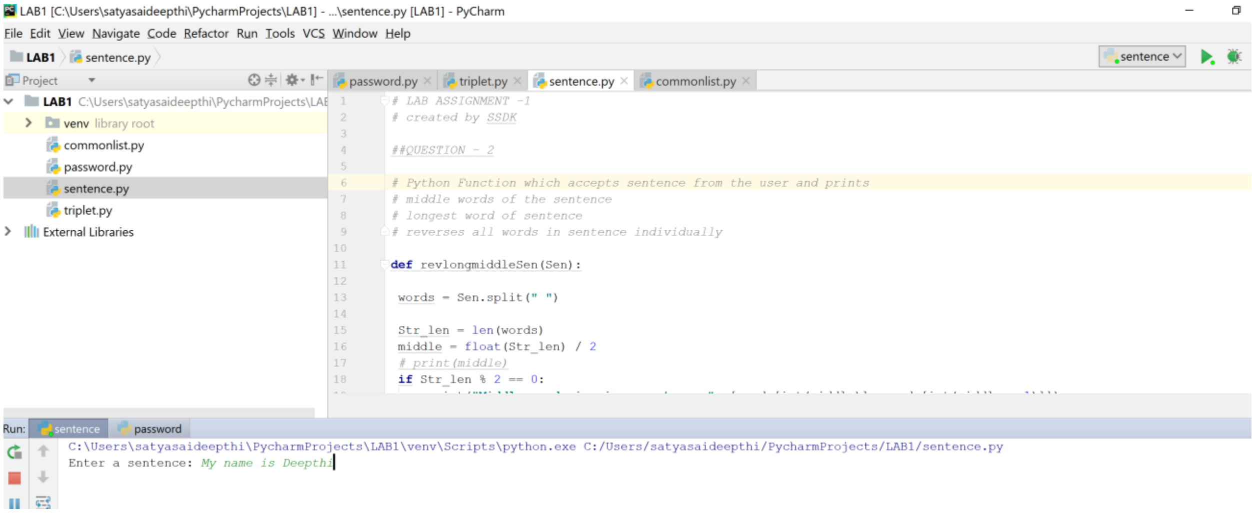

LAB1 [C:\Users\satyasaideepthi\PycharmProjects\LAB1] - ...\password.py [LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LAB1 password.py
password
Project
password.py triplet.py sentence.py commonlist.py
LAB1 C:\Users\satyasaideepthi\PycharmProjects\LAB
venv library root
commonlist.py
password.py
sentence.py
triplet.py
External Libraries
pswd_chk()
Run: password password

```
#LAB ASSIGNMENT -1
# created by SSDK

## QUESTION - 1

#function for password check
def pswd_chk():
 #if the given return value is true, while loop is executed!
 while True:
  in_pswd = input('Input Your Desired Password : ')
  n = len(in_pswd)
  if n < 6:
   print('Password length at least 6 letters')
  elif n > 16:
   print('Password length should not exceed 16 letters')
```

```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/password.py
Input Your Desired Password : deepu
Password length at least 6 letters
Input Your Desired Password : deepthi
Password must have at least one Numeric value
Input Your Desired Password : deepthi123
Password must have at least one Uppercase letter
Input Your Desired Password : Deepthi@123
Success! Your Password meets the requirements.
Your password is Deepthi@123

Process finished with exit code 0
```

<u>Task 2 Deployment:</u>

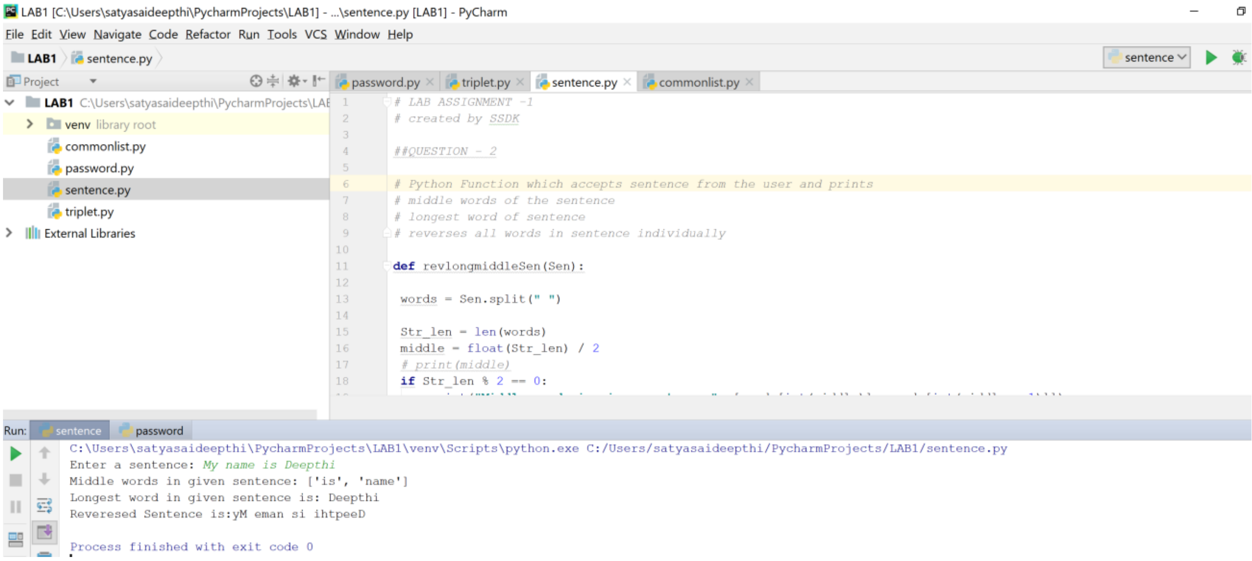
Step 1: User defined sentence is given as input.


LAB1 [C:\Users\satyasaideepthi\PycharmProjects\LAB1] - ...\sentence.py [LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LAB1 sentence.py
sentence
Project
password.py triplet.py sentence.py commonlist.py
LAB1 C:\Users\satyasaideepthi\PycharmProjects\LAB
venv library root
commonlist.py
password.py
sentence.py
triplet.py
External Libraries
Run: sentence password

```
# LAB ASSIGNMENT -1
# created by SSDK

##QUESTION - 2

# Python Function which accepts sentence from the user and prints
# middle words of the sentence
# longest word of sentence
# reverses all words in sentence individually

def revlongmiddleSen(Sen):

 words = Sen.split(" ")

 Str_len = len(words)
 middle = float(Str_len) / 2
 # print(middle)
 if Str_len % 2 == 0:
```

```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/sentence.py
Enter a sentence: My name is Deepthi
```

Step 2: The output for middle words, longest word and reversed sentence is displayed in console window.

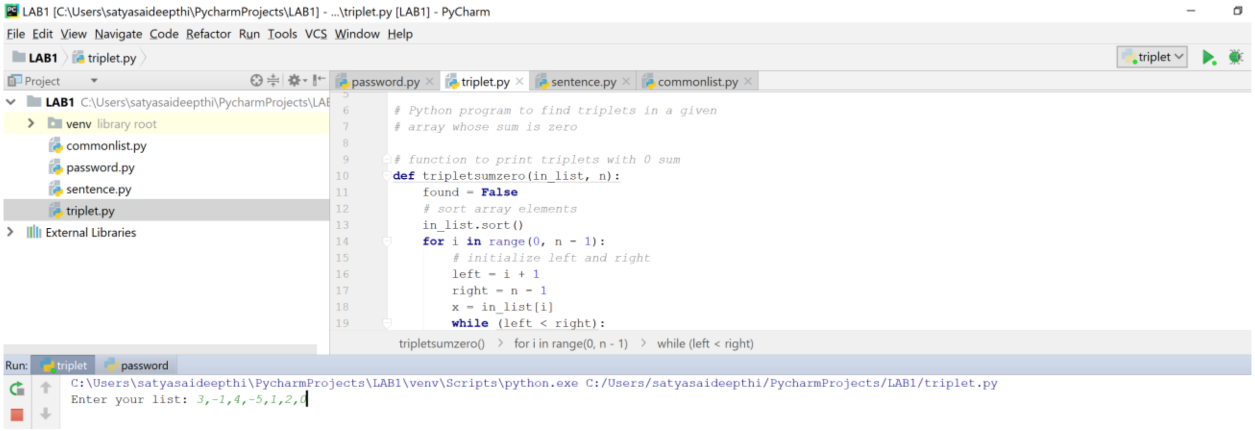

LAB1 [C:\Users\satyasaideepthi\PycharmProjects\LAB1] - ...\sentence.py [LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LAB1 sentence.py
sentence
Project
LAB1 C:\Users\satyasaideepthi\PycharmProjects\LAB
venv library root
commonlist.py
password.py
sentence.py
triplet.py
External Libraries
password.py triplet.py sentence.py commonlist.py
Run: sentence password

```
# LAB ASSIGNMENT -1
# created by SSDK

##QUESTION - 2

# Python Function which accepts sentence from the user and prints
# middle words of the sentence
# longest word of sentence
# reverses all words in sentence individually

def revlongmiddleSen(Sen):

 words = Sen.split(" ")

 Str_len = len(words)
 middle = float(Str_len) / 2
 # print(middle)
 if Str_len % 2 == 0:
```

```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/sentence.py
Enter a sentence: My name is Deepthi
Middle words in given sentence: ['is', 'name']
Longest word in given sentence is: Deepthi
Reveresed Sentence is:yM eman si ihtpeeD

Process finished with exit code 0
```

Task 3 Deployment:

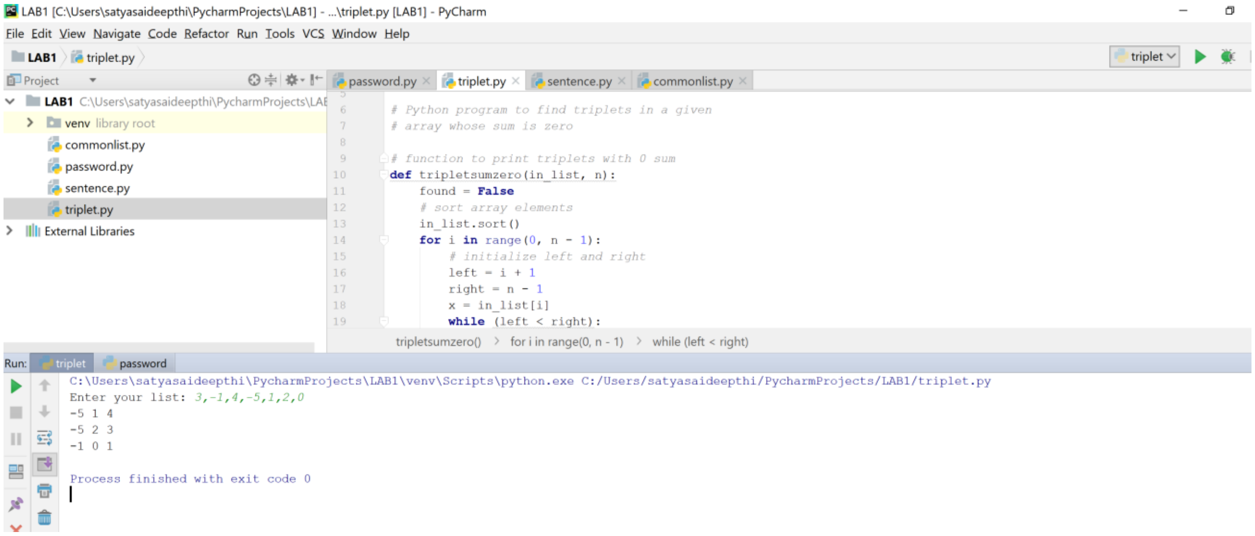
Step 1: Input the list of numbers separated by commas.


LAB1 [C:\Users\satyasaideepthi\PycharmProjects\LAB1] - ...\triplet.py [LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LAB1 triplet.py
triplet
Project
LAB1 C:\Users\satyasaideepthi\PycharmProjects\LAB
venv library root
commonlist.py
password.py
sentence.py
triplet.py
External Libraries
password.py triplet.py sentence.py commonlist.py
tripletsumzero() > for i in range(0, n - 1) > while (left < right)
Run: triplet password

```
# Python program to find triplets in a given
# array whose sum is zero

# function to print triplets with 0 sum
def tripletsumzero(in_list, n):
    found = False
    # sort array elements
    in_list.sort()
    for i in range(0, n - 1):
        # initialize left and right
        left = i + 1
        right = n - 1
        x = in_list[i]
        while (left < right):
```

```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/triplet.py
Enter your list: 3,-1,4,-5,1,2,0
```

Step 2: Result displaying triplet which sum to zero is taken as output.

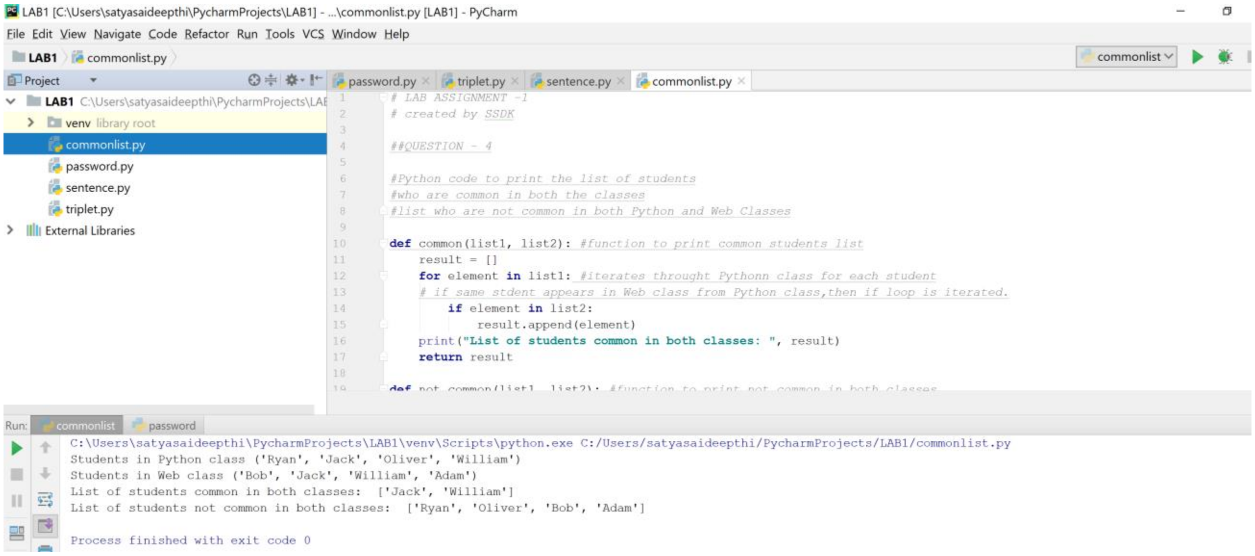

LAB1 [C:\Users\satyasaideepthi\PycharmProjects\LAB1] - ...\triplet.py [LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LAB1 triplet.py
triplet
Project
LAB1 C:\Users\satyasaideepthi\PycharmProjects\LAB
venv library root
commonlist.py
password.py
sentence.py
triplet.py
External Libraries
password.py triplet.py sentence.py commonlist.py
tripletsumzero() > for i in range(0, n - 1) > while (left < right)
Run: triplet password

```
# Python program to find triplets in a given
# array whose sum is zero

# function to print triplets with 0 sum
def tripletsumzero(in_list, n):
    found = False
    # sort array elements
    in_list.sort()
    for i in range(0, n - 1):
        # initialize left and right
        left = i + 1
        right = n - 1
        x = in_list[i]
        while (left < right):
```

```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/triplet.py
Enter your list: 3,-1,4,-5,1,2,0
-5 1 4
-5 2 3
-1 0 1

Process finished with exit code 0
```

Task 4 Deployment:

Step: As the input list is inbuilt in the code, the output is displayed in console printing output results along with given input.


LAB1 [C:\Users\satyasaideepthi\PycharmProjects\LAB1] - ...\commonlist.py [LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LAB1 commonlist.py
commonlist
Project
LAB1 C:\Users\satyasaideepthi\PycharmProjects\LAB
venv library root
commonlist.py
password.py
sentence.py
triplet.py
External Libraries
password.py triplet.py sentence.py commonlist.py
Run: commonlist password

```
# LAB ASSIGNMENT -1
# created by SSDK

##QUESTION - 4

#Python code to print the list of students
#who are common in both the classes
#list who are not common in both Python and Web Classes

def common(list1, list2): #function to print common students list
    result = []
    for element in list1: #iterates throught Pythonn class for each student
    # if same stdent appears in Web class from Python class,then if loop is iterated.
        if element in list2:
            result.append(element)
    print("List of students common in both classes: ", result)
    return result

def not_common(list1, list2): #function to print not common in both classes
```

```
C:\Users\satyasaideepthi\PycharmProjects\LAB1\venv\Scripts\python.exe C:/Users/satyasaideepthi/PycharmProjects/LAB1/commonlist.py
Students in Python class ('Ryan', 'Jack', 'Oliver', 'William')
Students in Web class ('Bob', 'Jack', 'William', 'Adam')
List of students common in both classes:  ['Jack', 'William']
List of students not common in both classes:  ['Ryan', 'Oliver', 'Bob', 'Adam']

Process finished with exit code 0
```

LIMITATIONS

The provided code meets all the requirements of the tasks given. But few limitations of the code are listed below:

- In Task 3, code doesn’t accept any float or double values and gives the triplet list only for integer values.
- In the final task, the input cannot be changed by user, a fixed list of students enrolled needs to be maintained all along the program to give the result.

REFERENCES

[1] https://stackoverflow.com/questions/41117733/validation-a-password-python

[2] https://stackoverflow.com/questions/32833575/how-to-find-the-longest-word-in-python

[3] https://www.geeksforgeeks.org/python-reverse-word-sentence/

[4] https://www.geeksforgeeks.org/find-triplets-array-whose-sum-equal-zero/

[5] https://stackoverflow.com/questions/2864842/common-elements-comparison-between-2-lists