Python Lab Assignment-1

Submitted by

Sravani Konujula

16230172/skv87

**Objective:**

The purpose of this assignment is

* To create a user password and its validation.
* To find middle words, longest word and reversing the sentence.
* Triplets of a set of numbers and respective sum must be equal to zero.
* To find common students and distinct students from two lists.

**Features:**

* It validates password weather it followed all the rules given and repeatedly runs if we gone some mistakes like if password is with 5 characters asks to retry password till it satisfies the condition.
* It is to find middle word in a sentence first it calculates the length of the sentence is odd prints middle word as the result if it is even prints the two middle words.
* And coming to longest word in a sentence it calculates the length of each word in a sentence and results the word with highest length.
* For the given set of numbers the triplets must be zero.
* From the both lists the common students resulted using intersection keyword and not common with union and difference keyword.

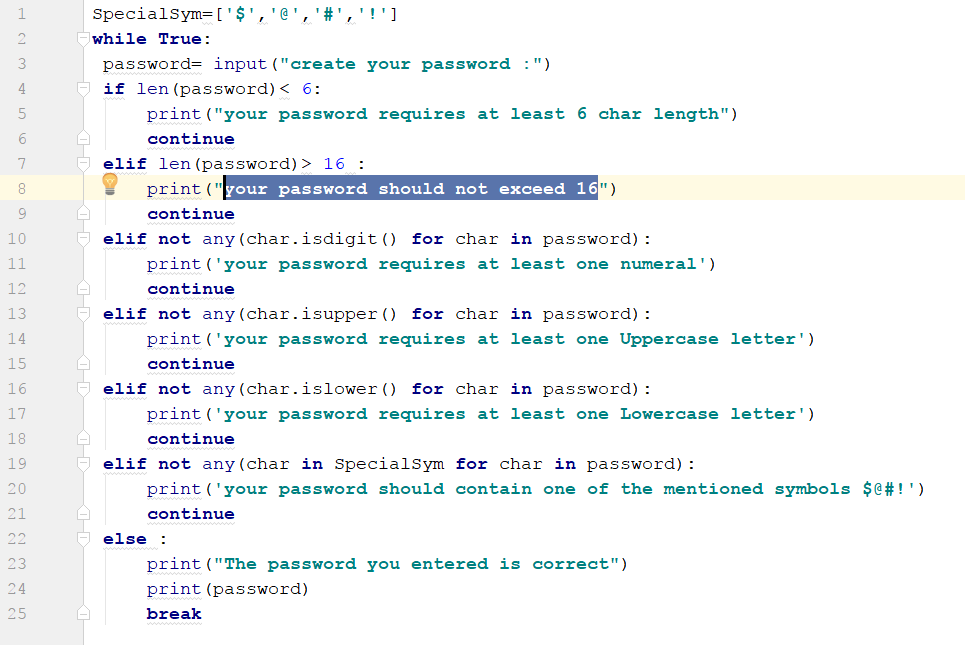
**Configuration:**

Pycharm

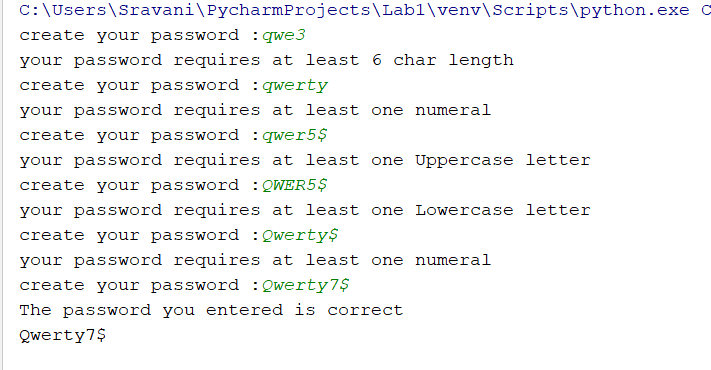
Python: 2.7.13

**Output Screens**

**Question-1**



**Output :-**

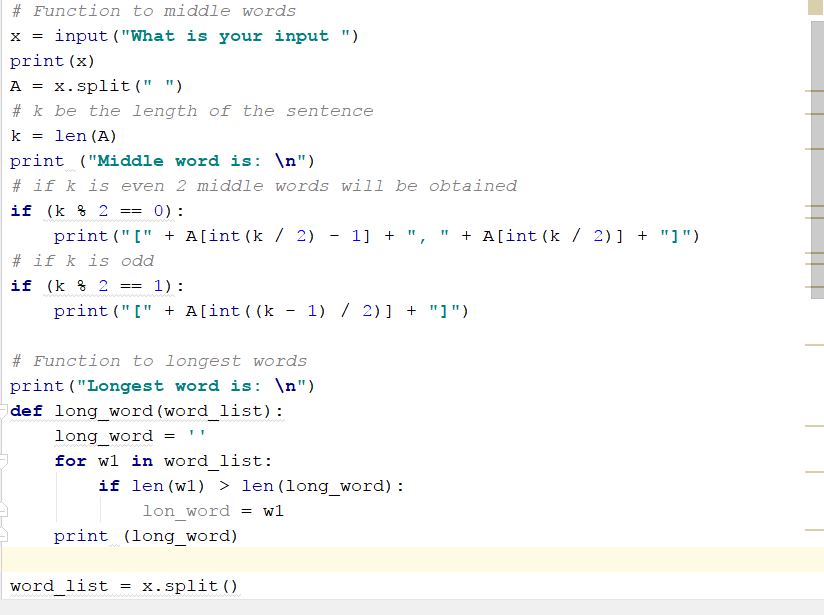


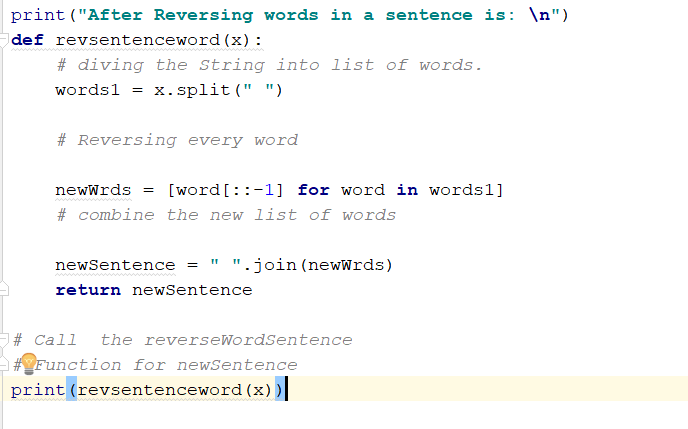
2) Write a Python function that accepts a sentence of words from user and display the following:

a) Middle word

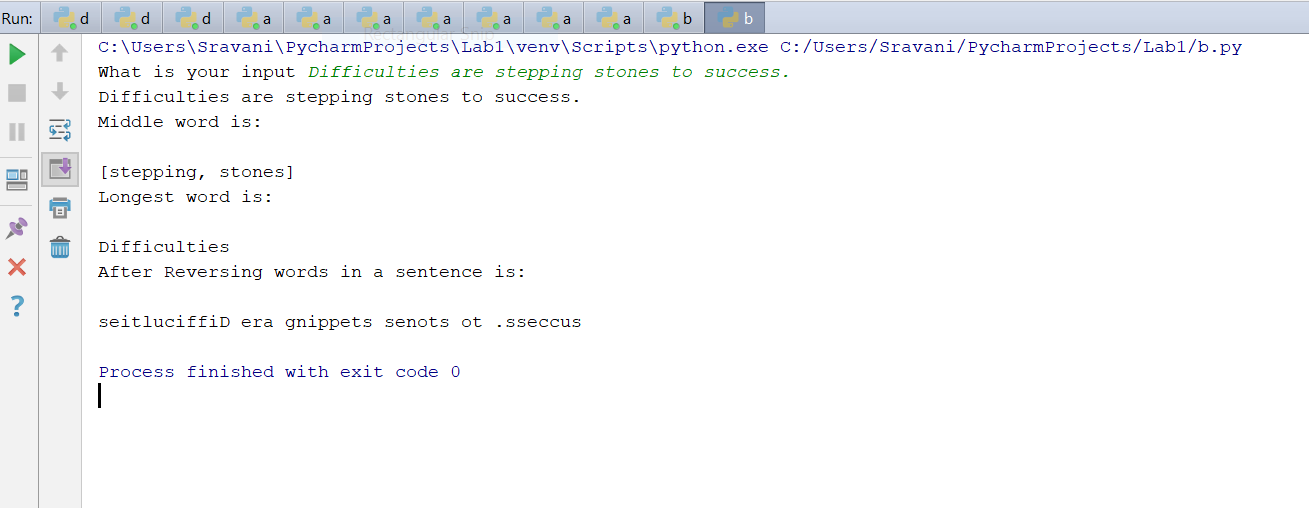
b) Longest word in the sentence

c) Reverse all the words in sentence





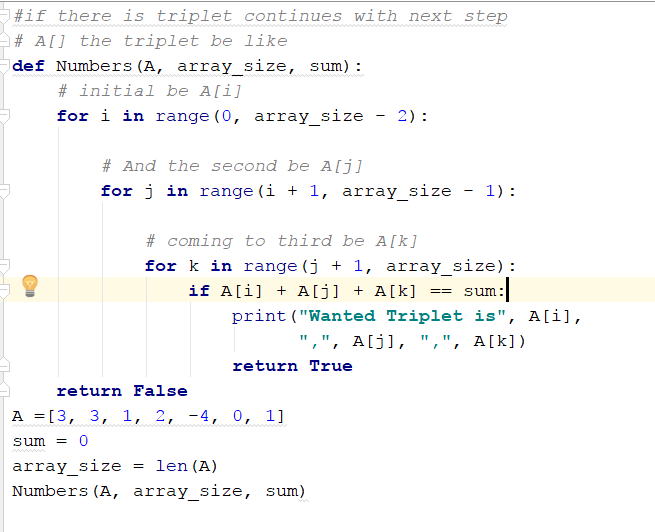
Output:



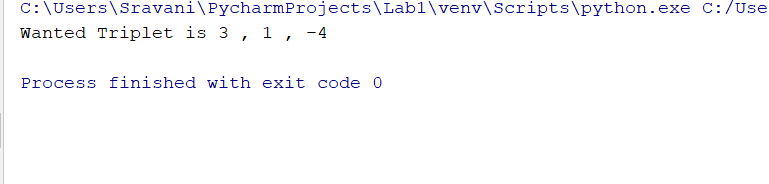
3) Given a list of n number, write a Python program to find triplets

in the list which gives the sum of zero.

**Source code :**



**Output:**



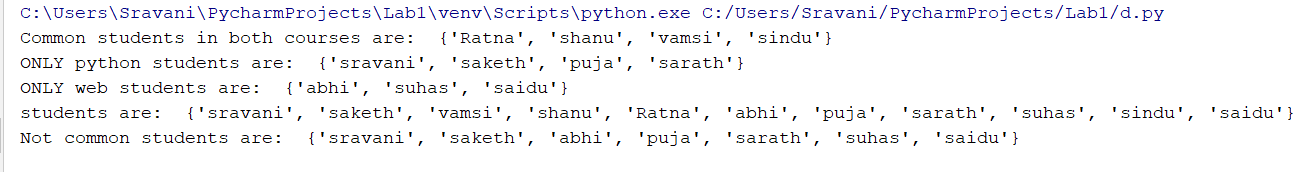
4) Consider the following scenario. You have a list of students who are attending class “Python” and another list of students who are attending class “Web Application”.

Find the list of students who are attending both the classes. Also find the list of students who are not common in both the classes. Print it.

**Source code :**



Output:



**Question1:**

**Code snippet1:**

In creating password we used char.isdigit(), char.isupper(), char.islower() and these methods are to satisfied.

SpecialSym=[**'$'**,**'@'**,**'#'**,**'!'**]  
**while True**:  
 password= input(**"create your password :"**)  
 **if** len(password)< 6:  
 print(**"your password requires at least 6 char length"**)  
 **continue  
 elif** len(password)> 16 :  
 print(**"your password should not exceed 16"**)  
 **continue  
 elif not** any(char.isdigit() **for** char **in** password):  
 print(**'your password requires at least one numeral'**)  
 **continue  
 elif not** any(char.isupper() **for** char **in** password):  
 print(**'your password requires at least one Uppercase letter'**)  
 **continue  
 elif not** any(char.islower() **for** char **in** password):  
 print(**'your password requires at least one Lowercase letter'**)  
 **continue  
 elif not** any(char **in** SpecialSym **for** char **in** password):  
 print(**'your password should contain one of the mentioned symbols $@#!'**)  
 **continue  
 else** :  
 print(**"The password you entered is correct"**)  
 print(password)  
 **break**

**Code snippet 2:**

Finding middle word in a sentence by using (k%2==0) and (k%2==1) and longest word is finding length of the word using “**def** find\_longest\_word” and reversing sentence is with reverseWordSentence.

*#* ***Function to middle words***x = input(**"What is your input "**)  
  
print(x)  
A = x.split(**" "**)  
*# k be the length of the sentence*k = len(A)  
print (**"Middle word is: \n"**)  
*# if k is even 2 middle words will be obtained***if** (k % 2 == 0):  
 print(**"["** + A[int(k / 2) - 1] + **", "** + A[int(k / 2)] + **"]"**)  
*# if k is odd***if** (k % 2 == 1):  
 print(**"["** + A[int((k - 1) / 2)] + **"]"**)

*#* ***Function to longest words***print(**"Longest word is: \n"**)  
**def** long\_word(word\_list):  
 long\_word = **''  
 for** w1 **in** word\_list:  
 **if** len(w1) > len(long\_word):  
 long\_word = w1  
 print (long\_word)  
  
word\_list = x.split()  
long\_word(word\_list)

*#* ***Function to Reverse words***print(**"After Reversing words in a sentence is: \n"**)  
**def** revsentenceword(x):  
 *# diving the String into list of words.* words1 = x.split(**" "**)  
  
 *# Reversing every word* newWrds = [word[::-1] **for** word **in** words1]  
 *# combine the new list of words* newSentence = **" "**.join(newWrds)  
 **return** newSentence  
  
*# Call the reverse Word Sentence  
# Function for newSentence*print(revsentenceword(x))

**code snippet3:**

*#if there is triplet continues with next step  
# A[] the triplet be like***def** Numbers(A, array\_size, sum):  
 *# initial be A[i]* **for** i **in** range(0, array\_size - 2):  
  
 *# And the second be A[j]* **for** j **in** range(i + 1, array\_size - 1):  
  
 *# coming to third be A[k]* **for** k **in** range(j + 1, array\_size):  
 **if** A[i] + A[j] + A[k] == sum:  
 print(**"Wanted Triplet is"**, A[i],  
 **","**, A[j], **","**, A[k])  
 **return True  
 return False**A =[3, 3, 1, 2, -4, 0, 1]  
sum = 0  
array\_size = len(A)  
Numbers(A, array\_size, sum)

**scode snippet4:**

Considering two list with python students and web students from that common students resulted using intersection keyword as “(set(py).intersection(set(web))))” and not common with “(set(py).union(set(web))).difference(set(py).intersection(set(web)))) “.

*#list of students enrolled in python class*py=[**'suhas'**,**'shanu'**,**'puja'**,**'saketh'**,**'sarath'**,**'vamsi'**,**'sravani'**,**'suddi'**]  
*#list of students enrolled in web class*web=[**'saidu'**,**'divya'**,**'abhi'**,**'shanu'**,**'vamsi'**,**'suhas'**,**'nam'**]  
  
*#to print students common in both classes  
#It is Used by sets Concept  
#Intersection is to extract common people.*print(**'Common students in both courses are: '**, set(py).intersection(set(web)))  
*#to print students Only in python class but in web*print(**'ONLY python students are: '**, set(py).difference(set(web)))  
*#to print students Only in web class but not in python*print(**'ONLY web students are: '**, set(web).difference(set(py)))  
*#Union is to find students in both web and python*print(**'students are: '**, set(py).union(set(web)))  
*#It is just the difference between union and intersection.*print(**'Not common students are: '**, (set(py).union(set(web))).difference(set(py).intersection(set(web))))

**Deployment:**

Code is written in python and we used pycharm to run this and printed result in the python console.

**Limitations:**

There are no limitations for the executed code snippets.

**References:**

<https://stackoverflow.com/questions/32833575/how-to-find-the-longest-word-in-python>

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

https://docs.python.org/2/library/sets.html