**LAB-1**

**PYTHON**

**AUTHOR:**

Harsha Saranam-16230289

**OBJECTIVES:**

1. To validate the password from user input by mentioned criteria.
2. To code the python function which results in printing the middle word, longest word and reversing the words in the sentence.
3. To find the triplets from the list whose sum is zero.
4. To find the list of students who are attending both the classes and who are not.

**FEATURES:**

1. Making sure the password matches the criteria

* Length should be between 6 and 16
* Checking whether the password contains a capital letter
* Checking whether the password contains a number
* Checking whether the password contains the special character

1. The output shows the middle word, longest word and each word in the sentence is reversed.
2. From the given set of numbers, the code finds the triplets whose sum is equal to zero and prints the set.
3. From the given lists, the code results the list of students who are common in both the classes and not.

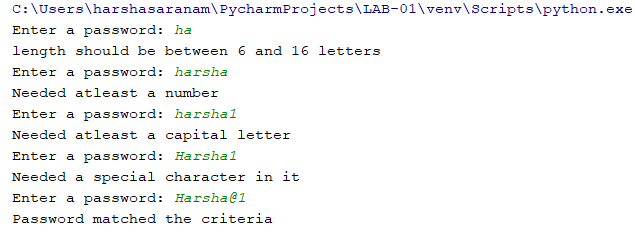
**CONFIGURATION:**

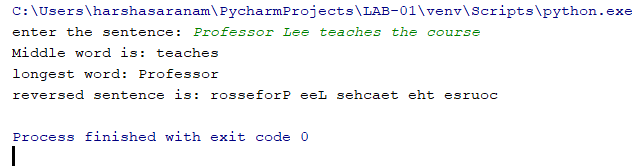
**Processor:** Intel Dual Core i7-6500U 2.5 GHz

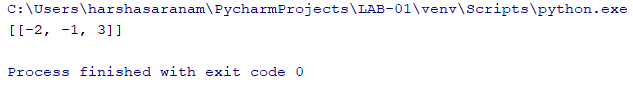
**RAM:** 8 GB

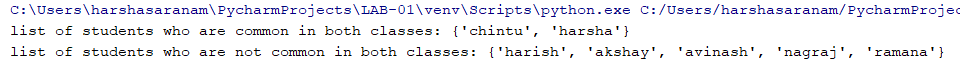
**SOFTWARE:** PYTHON IDE

**INPUT/OUTPUT:**









**IMPLEMENTATION:**

**QUESTION-01:**

**import** re *#this module is helpful for regular expressions matching operations***def** verify():  
 **while True**:  
 password = input(**"Enter a password: "**) *#inputting the password*

*#verifying the length of the input whether it matches the length criteria* **if** len(password) < 6 **or** len(password)>16:

print(**"length should be between 6 and 16 letters"**)

*#verifies whether it contains a number*

**elif** re.search(**'[0-9]'**,password) **is None**:

print(**"Needed atleast a number"**)

*#verifies whether it contains a capital letter*  
 **elif** re.search(**'[A-Z]'**,password) **is None**:

print(**"Needed atleast a capital letter"**)

*#verifies whether it contains a special character in it*  
 **elif** re.search(**'[[$@!\*]'**, password) **is None**:  
 print(**"Needed a special character in it"**) **else**:  
 print(**"Password matched the criteria"**)  
 **break**verify()

**QUESTION-02:**

sentence=input(**"enter the sentence: "**) *#inputing the sentence*final=sentence.split(**' '**) *#splitting the sentence with blank space stored in final*

revWords = [word[::-1] **for** word **in** final]  
newSentence = **" "**.join(revWords)

index=len(final) *#finding the length of the final*mid=int(index/2)  
**if** index==2: *#if the length is resulted as 2, we can conclude theres no middle word* print(**"no middle word"**)  
**if** index%2==0:*#if the length is even number, then the sentence contains 2 middle words* print(**"Middle words are:"**,final[mid-1:mid+1])  
**else** : *#if odd, prints the middle word, pointing with mid value* print(**"Middle word is:"**,final[mid])  
  
  
**def** findlong(final) : *#to find the longest in sentence* longest\_word=**' '** longest\_length=0  
 **for** word **in** final: *#for every word in final list, we r gonna find the longest* **if** len(word)>longest\_length:  
 longest\_word=word  
 longest\_length=len(word)  
  
 print(**"longest word:"**, longest\_word)  
findlong(final)  
  
print(**"reversed sentence is:"**, sentence[::-1]) *#printing the given sentence in reverse order*

**QUESTION-03:**

input = [1,3,6,2,-1,2,8,-2,9]  
output = []  
input.sort() *#sorting the list*r=len(input)-1  
**for** i **in** range(len(input)-2):  
 l = i + 1 *# we don't want l and i to be the same value.  
 # for each value of i, l starts one greater  
 # and increments from there.* **while** (l < r):  
 sum\_ = input[i] + input[l] + input[r]  
 **if** (sum\_ < 0):  
 l = l + 1  
 **if** (sum\_ > 0):  
 r = r - 1  
 **if not** sum\_: *# 0 is False in a boolean context* output.append([input[i],input[l],input[r]])  
 l = l + 1 *# increment l when we find a combination that works*print(output)

**QUESTION-04:**

pythonlist=[**'harsha'**,**'akshay'**,**'harish'**,**'avinash'**,**'chintu'**]*#list of students from python class*weblist=[**'harsha'**,**'chintu'**,**'nagraj'**] *#list of students from web class*

*#this function helps in finding the ppl who enrolled in both classes***def** common(a, b):

**return** set(a).intersection(set(b))

*#this function helps in finding the ppl who r not common in both classes***def** notcommon(a,b):

**return** (set(a).union(set(b)))-(set(a).intersection(set(b)))

print(**"list of students who are common in both classes:"**,common(pythonlist, weblist))  
print(**"list of students who are not common in both classes:"**,notcommon(pythonlist, weblist))

**REFERENCES:**

[**www.geeksforgeeks.com**](http://www.geeksforgeeks.com)

[**www.stackoverflow.com**](http://www.stackoverflow.com)