**Tool & Technique Laboratory (T&T Lab.)**

**[CS-3096]**

**Individual Work**

**Lab. No:4 , Date: 7/02/2023 , Day: 4**

**Topic: Python basics**

|  |  |  |  |
| --- | --- | --- | --- |
| **Roll Number:** | **20051133** | **Branch/Section:** | **CSE-17** |
| **Name in Capital:** | **Aryan Raj** | | |

**Program No: (**4.1)

**Original Program:**

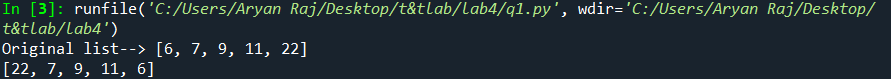
Python program to interchange first and second elements in a list.

**Modified Program Title:**

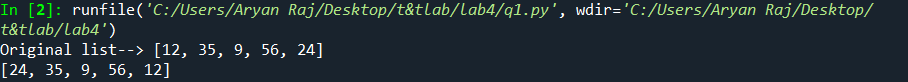
Python program to interchange first and last elements in a list.

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2**



**Source code**

def swapList(newList):

size = len(newList)

temp = newList[0]

newList[0] = newList[size - 1]

newList[size - 1] = temp

return newList

newList = [12, 35, 9, 56, 24]

print("Original list-->",newList)

print(swapList(newList))

**Conclusion/Observation**

We have successfully interchanged first and last number.

**Program No: (**4.2)

**Original Program:**

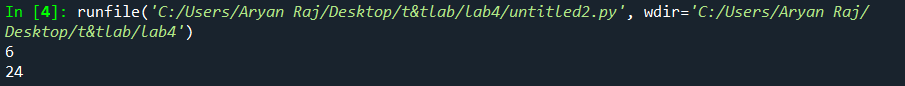
Sum of all numbers in the list.

**Modified Program Title:**

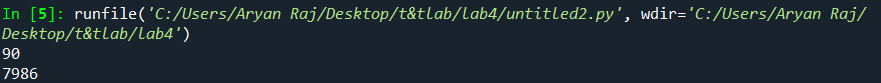
Multiply all numbers in the list

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2**



**Source code**

def multiplyList(myList):

result = 1

for x in myList:

result = result \* x

return result

list1 = [9, 2, 5]

list2 = [11, 22, 33]

print(multiplyList(list1))

print(multiplyList(list2))

**Conclusion/Observation**

We have successfully calculated product of all the numbers in the set.

**Program No: (**4.3)

**Original Program:**

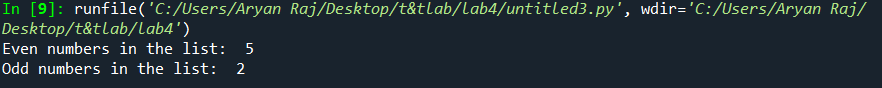
program to count Even numbers in a List.

**Modified Program Title:**

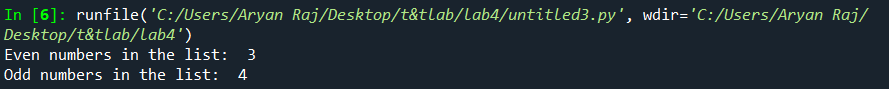
program to count Even and Odd numbers in a List

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2**



**Source code**

list1 = [9, 8, 4, 11, 6, 56, 2]

even\_count, odd\_count = 0, 0

for num in list1:

if num % 2 == 0:

even\_count += 1

else:

odd\_count += 1

print("Even numbers in the list: ", even\_count)

print("Odd numbers in the list: ", odd\_count)

**Conclusion/Observation**

We have successfully calculated number of odd and even numbers

**Program No: (**4.4)

**Original Program:**

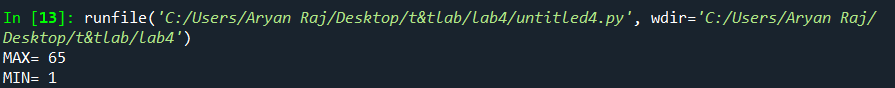
Maximum in a Set.

**Modified Program Title:**

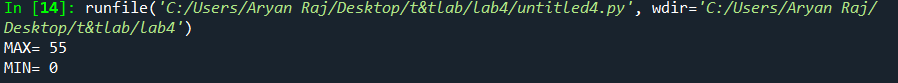
Maximum and Minimum in a Set.

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2**



**Source code**

sets = set([8, 16, 24, 1, 25, 3, 10, 65, 55])

print("MAX=",max(sets))

print("MIN=",min(sets))

**Conclusion/Observation**

We have successfully printed max and min number in a set.

**Program No: (**4.5)

**Original Program:**

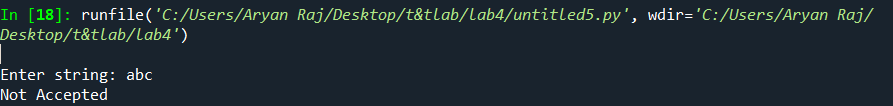
Program to accept a vowel.

**Modified Program Title:**

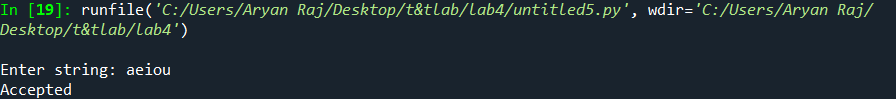
Program to accept the strings which contains all vowels

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2**



**Source code**

def check(string) :

string = string.lower()

vowels = set("aeiou")

s = set({})

for char in string :

if char in vowels :

s.add(char)

else:

pass

if len(s) == len(vowels) :

print("Accepted")

else :

print("Not Accepted")

if \_\_name\_\_ == "\_\_main\_\_" :

string = str(input("Enter string: "))

check(string)

**Conclusion/Observation**

Succefully accepted a string of all vowels

**Program No: (**4.6)

**Original Program:**

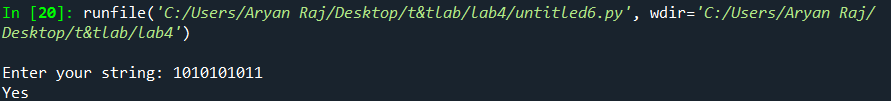
Check if a given string is containing any 0 or 1.

**Modified Program Title:**

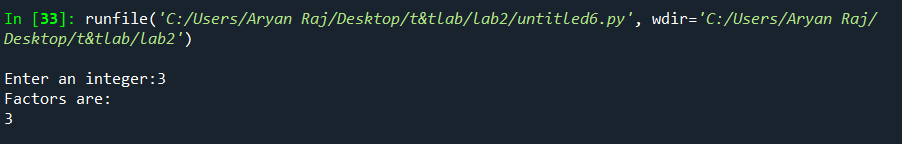
Check if a given string is binary string or not

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2**



**Source code**

def check(string):

p = set(string)

s = {'0', '1'}

if s == p or p == {'0'} or p == {'1'}:

print("Yes")

else:

print("No")

if \_\_name\_\_ == "\_\_main\_\_":

string = str(input("Enter your string: "))

check(string)

**Conclusion/Observation**

We have successfully checked if the string is binary or not.

**Program No: (**4.7)

**Original Program:**

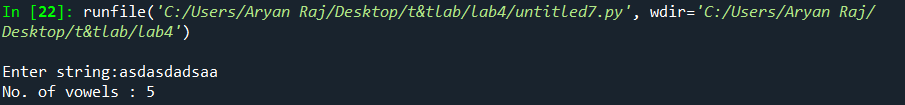
program to count number of alphabets using sets in a given string.

**Modified Program Title:**

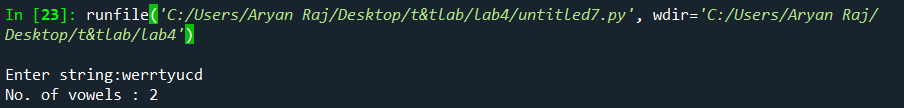
program to count number of vowels using sets in given string

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2**



**Source code**

def vowel\_count(str):

count = 0

vowel = set("aeiouAEIOU")

for alphabet in str:

if alphabet in vowel:

count = count + 1

print("No. of vowels :", count)

string = str(input("Enter string:"))

vowel\_count(string)

**Conclusion/Observation**

We have successfully counted total vowels

**Program No: (**4.8)

**Original Program:**

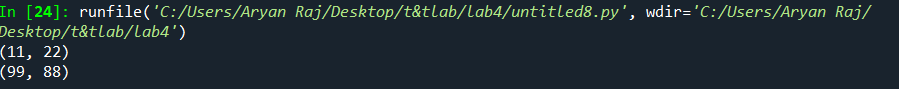
Swap two numbers in a tuple.

**Modified Program Title:**

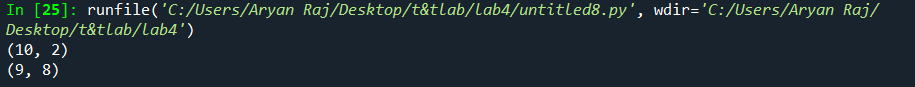
Swap two tuples in Python

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2:**



**Source code**

tuple1 = (10, 2)

tuple2 = (9, 8)

tuple1, tuple2 = tuple2, tuple1

print(tuple2)

print(tuple1)

**Conclusion/Observation**

We have successfully swapped two tupples.

**Program No: (**4.9)

**Original Program:**

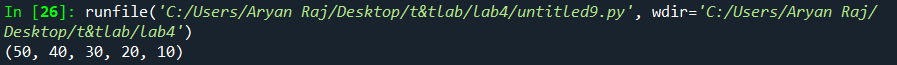
Print the tuple

**Modified Program Title:**

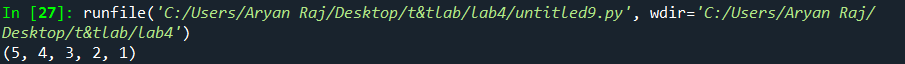
Reverse the tuple

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2:**



**Source code**

tuple1 = (1, 2, 3, 4, 5)

tuple1 = tuple1[::-1]

print(tuple1)

**Conclusion/Observation**

We have successfully reversed the tuple

**Program No: (**4.10)

**Original Program:**

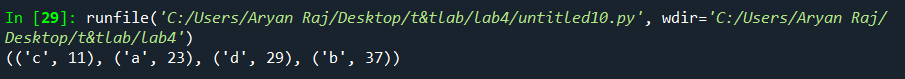
Sort the tuple

**Modified Program Title:**

Sort a tuple of tuples by 2nd item

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2:**



**Source code**

tuple1 = (('a', 2), ('b', 7), ('c', 1), ('d', 9))

tuple1 = tuple(sorted(list(tuple1), key=lambda x: x[1]))

print(tuple1)

**Conclusion/Observation**

We have sorted in the required order.

**Program No: (**4.11)

**Original Program:**

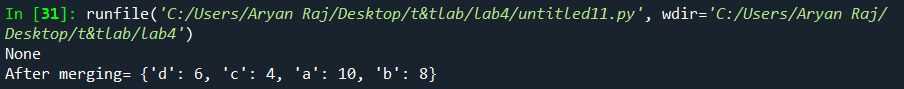
Print two Dictionaries in sorted order

**Modified Program Title:**

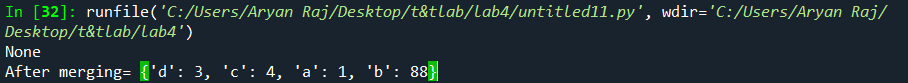
Merging two Dictionaries

**Input/Output Screenshots:**

**RUN-1:**



**RUN-2:**



**Source code**

def Merge(dict1, dict2):

return(dict2.update(dict1))

dict1 = {'a': 1, 'b': 88}

dict2 = {'d': 3, 'c': 4}

print(Merge(dict1, dict2))

print("After merging=" ,dict2)

**Conclusion/Observation**

We have merged two dictionaries.

**Program No: (**4.12)

**Original Program:**

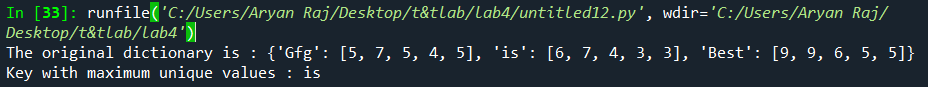
Print all keys

**Modified Program Title:**

Key with maximum unique values

**Input/Output Screenshots:**

**RUN-1:**



**Source code**

test\_dict = {"Gfg" : [5, 7, 5, 4, 5],

"is" : [6, 7, 4, 3, 3],

"Best" : [9, 9, 6, 5, 5]}

print("The original dictionary is : " + str(test\_dict))

max\_val = 0

max\_key = None

for sub in test\_dict:

if len(set(test\_dict[sub])) > max\_val:

max\_val = len(set(test\_dict[sub]))

max\_key = sub

print("Key with maximum unique values : " + str(max\_key))

**Conclusion/Observation**

We have printed keys with max unique values.

**Program No: (**4.13)

**Original Program:**

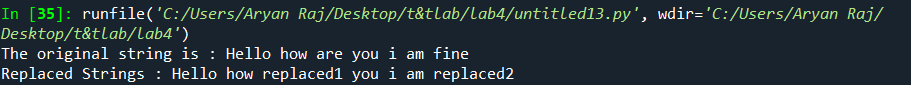
Remove words from Dictionary

**Modified Program Title:**

Replace words from Dictionary

**Input/Output Screenshots:**

**RUN-1:**



**Source code**

test\_str = 'Hello how are you i am fine'

print("The original string is : " + str(test\_str))

lookp\_dict = {"are" : "replaced1", "fine" : "replaced2"}

temp = test\_str.split()

res = []

for wrd in temp:

res.append(lookp\_dict.get(wrd, wrd))

res = ' '.join(res)

print("Replaced Strings : " + str(res))

**Conclusion/Observation**

We have replaced values from dictionary