**Cycle II**

**program1**

**Aim:-**

Create a string from the given string where the first and last character are exchanged.

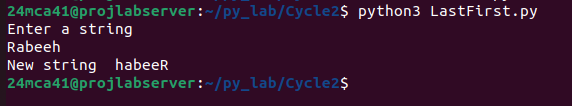
Eg: Python ⇒ nythoP

**Source code:-**

str = input("Enter a string\n")

newString = str[-1] + str[1:-1] + str[0]

print("New string ", newString)



**program2**

**Aim:-**

Get a string from an input string where all occurrences of the first character are

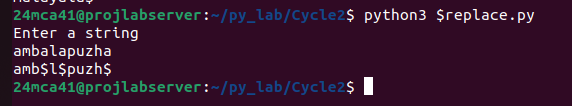
replaced with ‘$’, except the first character. [eg: onion -> oni$n]

**Source code:-**

str = input("Enter a string\n")

newString = str[0] + str[1:].replace(str[0], '$')

print(newString)



**program3**

**Aim:-**

Create a single string separated with space from two strings by swapping the character

at position 1.

Eg : str1 = “Hello” str2 =”World” , then create a string str3 = “Hollo Werld” [Hint: use

slicing and concatenation ]

**Source code:-**

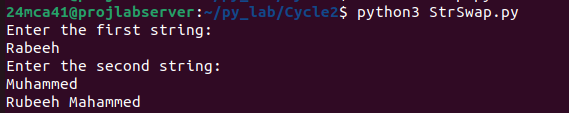
strOne = input("Enter the first string\n")

strTwo = input("Enter the second string\n")

newStr1 = strOne[0] + strTwo[1:2] + strOne[2:]

newStr2 = strTwo[0] + strOne[1:2] + strTwo[2:]

print(newStr1 + " " + newStr2)



**program4**

**Aim:-**

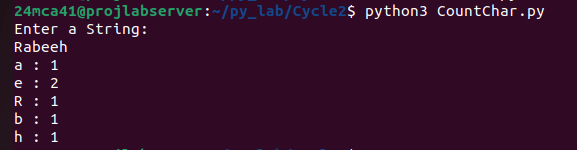
Count the number of characters (character frequency) in a string

**Source code:-**

String1=input("Enter a String:\n")

for Char in set(String1):

print(Char,":",String1.count(Char))

****

**program5**

**Aim:-**

Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’

**Source code:-**

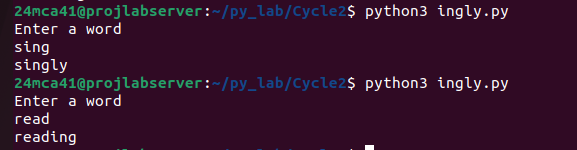
str = input("Enter a word\n")

if(str[-3:]) == "ing":

print(str + "ly")

else:

print(str + "ing")

****

**program6**

**Aim:-**

Store a list of first names. Count the occurrences of ‘a’ within the list

**Source code:-**

names=["Amal","Rabeeh","Shreyas","Abhijith"]

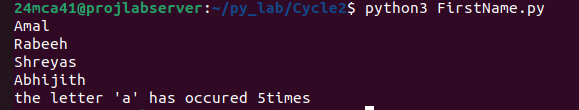
a\_count=0;

for name in names:

print(name)

a\_count=a\_count+name.lower().count("a")

print(f"the letter 'a' has occured {a\_count}times")

****

**program7**

**Aim:-**

Write a python program to read two lists color-list1 and color-list2. Print out all colors

from color-list1 not contained in color-list2.

**Source code:-**

list1=input("Enter colors seperated by commas:")

l ist2=input("Enter list2 colors seperated by commmas:")

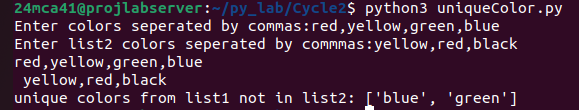
print(list1,"\n",list2)

set1=set(list1.split(','))

set2=set(list2.split(','))

result=[color for color in set1 if color not in set2]

print("unique colors from list1 not in list2:",result)

****

**program8**

**Aim:-**

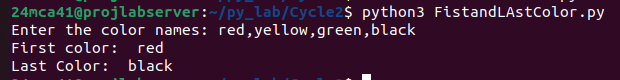
Create a list of colors from comma-separated color names entered by the user. Display

first and last colors.

**Source code:-**

colors=input("Enter the color names: ").split(',')

print("First color: ",colors[0],"\nLast Color: ",colors[-1])

****

**program9**

**Aim:-**

Write a program to prompt the user for a list of integers. For all values greater than

100,store ‘over’ instead

**Source code:-**

Integer1=input("enter integers to list: ").split(',')

for i in range(len(Integer1)):

if int(Integer1[i])>100:

Integer1[i]='over'

print("List: ",Integer1)

****

**program10**

**Aim:-**

From a list of integers, create a list after removing even numbers.

**Source code:-**

Integer1=list(map(int,input("enter integers to list seperated by spaces: ").split()))

print("List before Removal: ",Integer1)

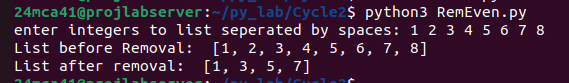
result=[]

for i in Integer1:

if i%2==0:

Integer1.remove(i)

print("List after removal: ",Integer1)

****

**program11**

**Aim:-**

Accept a list of words and return the length of the longest word.

**Source code:-**

words=input("Enter words to find length: ").split()

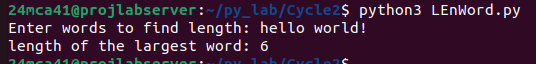
if not words:

print("List is empty:")

else:

greater=len(max(words, key=len))

print("length of the largest word:",greater)

****

**program12**

**Aim:-**

Write a program to prompt the user to enter two lists of integers and check

(a) Whether lists are of the same length.

(b) Whether the list sums to the same value.

(c) Whether any value occurs in both Lists

**Source code:-**

list1=list(map(int,input("enter list1: ").split()))

list2=list(map(int,input("Enter list2: ").split()))

same=[]

if not list1 or not list2:

print("Length are same: ",len(list1))

else:

print("length are not same")

for i in list1:

if i in list2 and i not in same:

same.append(i)

sum1=sum(list1)

sum2=sum(list2)

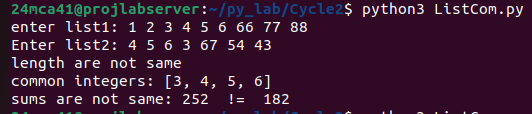
print("common integers:",same)

if sum1==sum2:

print("sums are same:",sum1,"=",sum2)

else:

print("sums are not same:",sum1," != ",sum2)

****

**program13**

**Aim:-**

Write a Python program to count the occurrences of each word in a line of text.

Hint: use split() function and dictionary

Sample input : the quick brown fox jumps over the lazy dog

Output : {'the': 2, 'jumps': 1, 'brown': 1, 'lazy': 1, 'fox': 1, 'over': 1, 'quick': 1, 'dog.': 1}

**Source code:-**

words=input("Enter String using spaces: ").split()

word\_count={}

for word in words:

words=word.lower()

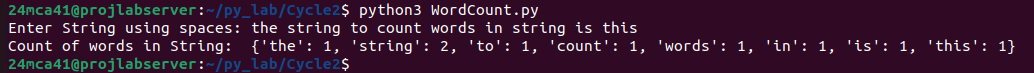
if word in word\_count:

word\_count[word]+=1

else:

word\_count[word]=1

print("Count of words in String: ",word\_count)

****

**program14**

**Aim:-**

List comprehensions:

(a) Generate positive list of numbers from a given list of integers

(b) Square of N numbers

(c) Form a list of vowels selected from a given word

(d) Form a list ordinal value of each element of a word (Hint: use ord() to get

ordinal values)

**Source code:-**

numbers=list(map(int,input("Enter Integers: ").split()))

positive\_numbers=[num for num in numbers if num>0]

print("positie numers: ",positive\_numbers)

N=int(input("input N: "))

squares=[num \*\*2 for num in range(1,N+1)]

print("squares of ",N," :",squares)

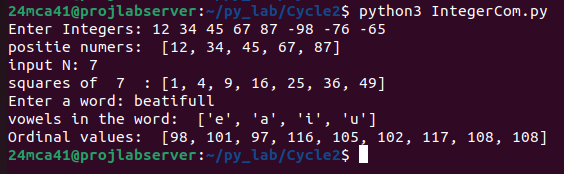
word=input("Enter a word: ")

vowel=[char for char in word if char in 'aeiouAEIOU']

print("vowels in the word: ",vowel)

ordinal\_values=[ord(char) for char in word]

print("Ordinal values: ",ordinal\_values)



**program15**

**Aim:-**

Sort dictionary in ascending and descending order

**Source code:-**

user\_dict={}

n=int(input("Enter the number of key-value pair: "))

for \_ in range(n):

key=input("Enter Key: ")

value=int(input("Enter value: "))

user\_dict[key]=value

asc\_by\_keys=dict(sorted(user\_dict.items()))

desc\_by\_keys=dict(sorted(user\_dict.items(),reverse=True))

asc\_by\_values=dict(sorted(user\_dict.items(), key=lambda items:items[1]))

desc\_by\_values=dict(sorted(user\_dict.items(), key=lambda items:items[1], reverse=True))

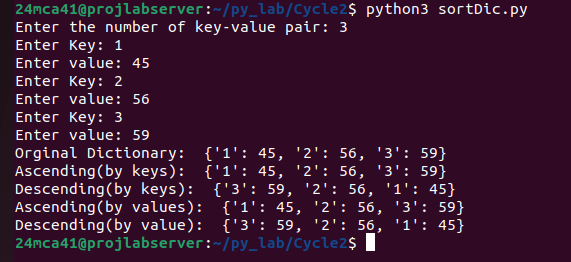
print("Orginal Dictionary: ",user\_dict)

print("Ascending(by keys): ",asc\_by\_keys)

print("Descending(by keys): ",desc\_by\_keys)

print("Ascending(by values): ",asc\_by\_values)

print("Descending(by value): ",desc\_by\_values)



**program16**

**Aim:-**

**Merge two dictionaries.**

**Source code:-**

def input\_dict(n):

user\_dict={}

for \_ in range(n):

key=input("Enter Key: ")

value=int(input("Enter Value; "))

user\_dict[key]=value

return user\_dict

n1=int(input("Enter number of key value for dict1: "))

dict1=input\_dict(n1)

n2=int(input("Enter number of key value for dict2: "))

dict2=input\_dict(n2)

dict1.update(dict2)

print("Merged dictionary:",dict1)

