1. WAP to find the length of string

def string\_length(str1):

count = 0

for char in str1:

count += 1

return count

print(string\_length('softwarica'))

1. WAP to find the character frequency from given string.

def char\_frequency(str1):

dict = {}

for n in str1:

keys = dict.keys()

if n in keys:

dict[n] += 1

else:

dict[n] = 1

return dict

print(char\_frequency('google.com'))

1. WAP that takes a list of words and find the length of longest one.

def find\_longest\_word(words\_list):

word\_len = []

for n in words\_list:

word\_len.append((len(n), n))

word\_len.sort()

return word\_len[-1][1]

print(find\_longest\_word(["PHP", "Exercises", "Backend"]))

1. WAP to remove odd index characters from the given string.

def odd\_values\_string(str):

result = ""

for i in range(len(str)):

if i % 2 == 0:

result = result + str[i]

return result

print(odd\_values\_string('abcdef'))

print(odd\_values\_string('python'))

1. WAP to print even length words from given string.

# print EVEN length words of a string

# declare, assign string

str = "Python is a programming language"

# extract words in list

words = list(str.split(' '))

# print string

**print** "str: ", str

# print list converted string i.e. list of words

**print** "list converted string: ", words

# iterate words, get length

# if length is EVEN print word

**print** "EVEN length words:"

**for** W **in** words:

**if**(len(W)%2==0 ):

**print** W

1. Count vowels from the given string

# count vowels in a string

# declare, assign string

str = "Hello world"

# declare count

count = 0

# iterate and check each character

**for** i **in** str:

# check the conditions for vowels

**if**( i=='A' **or** i=='a' **or** i=='E' **or** i=='e'

**or** i=='I' **or** i=='i' **or** i=='O' **or** i=='o'

**or** i=='U' **or** i=='u'):

count +=1;

# print count

**print** "Total vowels are: ", coun

1. WAP to check if substring exists in a given string.

# python program to check substring

# presents in the string or not

# string and substring declaration, initialization

str = "IncludeHelp.Com"

sub\_str ="Help"

# checking sub\_str presents in str or not

**if** sub\_str **in** str:

**print**("Yes, substring presents in the string.")

**else**:

**print**("No, substring does not present in thestring.");

# testing another substring

sub\_str = "Hello"

# checking sub\_str presents in str or not

**if** sub\_str **in** str:

**print**("Yes, substring presents in the string.")

**else**:

**print**("No, substring does not present in thestring.");

1. Print double quotes with string variable

#declare a string

str1 = "Hello world";

#printing string with the double quotes

**print**("\"%s\"" % str1)

**print**('"%s"' % str1)

**print**('"{}"'.format(str1))

1. WAP to reverse the given string.

string = input('Enter a string : ')

# reverse a string using string slicing concept

rev\_string = string[::-1]

**print**("reverse string :",rev\_string)

or

string = input('Enter a string : ')

length = len(string)

rev\_string = ''

# iteration from the last character till

# first character and cocatenating them

**for** index **in** range(length-1,-1,-1) :

rev\_string += string[index]

**print**("reverse string :",rev\_string)

or

# define a function for reversing the string

**def** reverseString(string) :

length = len(string)

rev\_string = ''

# iteration from the last character till

# first character and cocatenating them

**for** index **in** range(length-1,-1,-1) :

rev\_string += string[index]

**return** rev\_string

# Main() method

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

string = input('Enter a string : ')

**print**("reverse string :",reverseString(string))