**Python Programming Basic Assignment-25**

**Question1**

Create a function that takes three integer arguments (a, b, c) and returns the amount of integers which are of equal value.

Examples

equal(3, 4, 3) ➞ 2

equal(1, 1, 1) ➞ 3

equal(3, 4, 1) ➞ 0

Notes

Your function must return 0, 2 or 3.

**def** equal(**\***args):

new**=**[]

**for** i **in** args:

c**=**args**.**count(i)

**if** c**>**1:

new**.**append(i)

**return** len(new)

equal(3,4,3)

2

In [24]:

equal(1, 1, 1)

Out[24]:

3

In [25]:

equal(3, 4, 1)

Out[25]:

0

**Question2**

Write a function that converts a dictionary into a list of keys-values tuples.

Examples

dict\_to\_list({ "D": 1, "B": 2, "C": 3 }) ➞ [("B", 2), ("C", 3), ("D", 1)]

dict\_to\_list({ "likes": 2, "dislikes": 3, "followers": 10 }) ➞ [("dislikes", 3), ("followers", 10), ("likes", 2)]

Notes Return the elements in the list in alphabetical order.

In [35]:

**def** dict\_to\_list(d):

sorted\_list**=**sorted(d**.**items())

print(list(sorted\_list))

In [36]:

dict\_to\_list({ "D": 1, "B": 2, "C": 3 })

[('B', 2), ('C', 3), ('D', 1)]

In [37]:

dict\_to\_list({ "likes": 2, "dislikes": 3, "followers": 10 })

[('dislikes', 3), ('followers', 10), ('likes', 2)]

**Question3**

Write a function that creates a dictionary with each (key, value) pair being the (lower case, upper case) versions of a letter, respectively.

Examples

mapping(["p", "s"]) ➞ { "p": "P", "s": "S" }

mapping(["a", "b", "c"]) ➞ { "a": "A", "b": "B", "c": "C" }

mapping(["a", "v", "y", "z"]) ➞ { "a": "A", "v": "V", "y": "Y", "z": "Z" }

Notes

All of the letters in the input list will always be lowercase.

In [38]:

**def** mapping(l):

d**=**{}

**for** i **in** l:

d[i]**=**i**.**upper()

**return** d

In [48]:

mapping(['p','s'])

Out[48]:

{'p': 'P', 's': 'S'}

In [49]:

mapping(["a", "b", "c"])

Out[49]:

{'a': 'A', 'b': 'B', 'c': 'C'}

In [50]:

mapping(["a", "v", "y", "z"])

Out[50]:

{'a': 'A', 'v': 'V', 'y': 'Y', 'z': 'Z'}

**Question4**

Write a function, that replaces all vowels in a string with a specified vowel.

Examples

vow\_replace("apples and bananas", "u") ➞ "upplus und bununus"

vow\_replace("cheese casserole", "o") ➞ "chooso cossorolo"

vow\_replace("stuffed jalapeno poppers", "e") ➞ "steffed jelepene peppers"

Notes

All words will be lowercase. Y is not considered a vowel.

In [81]:

**def** vow\_replace(string,char):

vow**=**'AEIOUaeiou'

**for** i **in** vow:

string**=**string**.**replace(i,char)

**return** string

In [82]:

vow\_replace("apples and bananas", "u")

Out[82]:

'upplus und bununus'

In [83]:

vow\_replace("cheese casserole", "o")

Out[83]:

'chooso cossorolo'

In [84]:

vow\_replace("stuffed jalapeno poppers", "e")

Out[84]:

'steffed jelepene peppers'

**Question5**

Create a function that takes a string as input and capitalizes a letter if its ASCII code is even and returns its lower case version if its ASCII code is odd.

Examples

ascii\_capitalize("to be or not to be!") ➞ "To Be oR NoT To Be!"

ascii\_capitalize("THE LITTLE MERMAID") ➞ "THe LiTTLe meRmaiD"

ascii\_capitalize("Oh what a beautiful morning.") ➞ "oH wHaT a BeauTiFuL moRNiNg."

In [101]:

**def** ascii\_capitalize(string):

**for** i **in** string:

**if** ord(i)**%2**==0:

string**=**string**.**replace(i,i**.**upper())

**else**:

string**=**string**.**replace(i,i**.**lower())

**return** string

In [102]:

ascii\_capitalize("to be or not to be!")

Out[102]:

'To Be oR NoT To Be!'

In [103]:

ascii\_capitalize("THE LITTLE MERMAID")

Out[103]:

'THe LiTTLe meRmaiD'

In [104]:

ascii\_capitalize("Oh what a beautiful morning.")

Out[104]:

'oH wHaT a BeauTiFuL moRNiNg.'