**BASIC PROGRAMMING ASSIGNMENT\_25 -SUBMITTED BY SAMUEL DEVDAS**

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.

Ans.

def equal(a,b,c):

args=[a,b,c]

set\_args=set(args)

if len(set\_args)==1:

return(3)

elif len(set\_args)==3:

return(0)

elif len(set\_args)==2:

return(2)

equal(3, 4, 1)

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.

Ans.

def dict\_to\_list(dict):

l=dict.items()

return sorted(list(l))

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

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.

Ans.

def mapping(lst):

dict={}

for elem in lst:

dict[elem]=elem.capitalize()

return dict

mapping(["a", "v", "y", "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.

Ans.

def vow\_replace(string,v):

vowels=['a','e','i','o','u']

replaced=[]

for char in string:

if char in vowels:

replaced.append(v)

else:

replaced.append(char)

replaced\_str=''

for elem in replaced:

replaced\_str+=elem

return (replaced\_str)

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

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."

Ans.

def ascii\_capitalize(string):

caps=[]

for char in string:

if ord(char)%2==0:

caps.append(char.capitalize())

else:

caps.append(char.lower())

cap\_odd\_ascii=''

for elem in caps:

cap\_odd\_ascii+=elem

return(cap\_odd\_ascii)

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