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(a, b, c) :

num = 0

if a == b and a == c :

num = 3

elif a == b or a == c :

num = 2

return num

print(equal(3, 4, 3))

print(equal(1, 1, 1))

print(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.

def dict\_to\_list1(dict1):

l=[]

for key,values in dict1.items():

l.append((key,values))

return l

print(dict\_to\_list1({"D" : 1, "B": 2, "C" : 3} ))

print(dict\_to\_list1({"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.

def mapping(in\_list):

out\_dict = {}

for ele in in\_list:

out\_dict[ele] = ele.upper()

print(out\_dict)

print(mapping(["p","s"]))

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

print(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.

def vow\_replace(str1,k):

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

out\_string = ''

for i in str1:

if i in vowels:

out\_string+=k

else:

out\_string += i

return out\_string

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

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."

def ascii\_capitalize(in\_string):

out\_string = ''

for i in in\_string.lower():

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

out\_string+=i.upper()

else:

out\_string+=i

return out\_string

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

print(ascii\_capitalize("THE LITTLE MERMAID"))