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.

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.

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.

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.

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."

**Solution: 1**

def equal(a, b, c):

li\_new = []

li\_new.append(a)

li\_new.append(b)

li\_new.append(c)

li\_check = []

for i in li\_new:

if i not in li\_check and li\_new.count(i)>1:

li\_check.append(li\_new.count(i))

if(li\_check.\_\_len\_\_()==0):

return(0)

else:

return(li\_check[0])

equal(3,1,1)

**Solution: 2**

def dic\_to\_list(d):

a = [((ke), val) for ke, val in sorted(d.items())]

return a

dic\_to\_list({

"D": 1,

"B": 2,

"C": 3

})

**Solution: 3**

def mapping(list\_given):

res\_dict = {list\_given[i]:list\_given[i].upper() for i in range(0, len(list\_given))}

return res\_dict

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

**Solution: 4**

def vow\_replace(tc, specified\_vowel):

vowels = "aeiouAEIOU"

for i in vowels:

tc = tc.replace(i, specified\_vowel)

return(tc)

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

**Solution: 5**

def ascii\_capitalize(sente):

for i in sente:

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

print(i.upper())

else:

print(i.lower())

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