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):

# If all three numbers are equal

if a == b == c:

return 3

# If two numbers are equal

elif a == b or b == c or a == c:

return 2

# If all numbers are different

else:

return 0

print(equal(3, 4, 3)) #2

print(equal(1, 1, 1)) #3

print(equal(3, 4, 1)) #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.

def dict\_to\_list(n):

# Convert the dictionary to a list of key-value pairs sorted by key

result = sorted(n.items())

# Print each key-value pair

for i, j in result:

print(list((i, j)))

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)]

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(n):

d = {}

for i in n:

d[i] = i.upper()

print(d)

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

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

mapping(["a", "v", "y", "z"]) #{ "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.

def vow\_replace(m, n):

vowels = 'aeiou' # Simplify the list of vowels into a string

for vowel in vowels:

# Replace all occurrences of the vowel with the provided character `n`

m = m.replace(vowel, n)

print(m)

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"

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(n):

result = ""

for i in n:

if ord(i) % 2 == 0:

result += i.capitalize()

else:

result += i.lower()

print(result)

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."