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({
&quot;D&quot;: 1,
&quot;B&quot;: 2,
&quot;C&quot;: 3
}) ➞ [(&quot;B&quot;, 2), (&quot;C&quot;, 3), (&quot;D&quot;, 1)]
dict_to_list({
&quot;likes&quot;: 2,
&quot;dislikes&quot;: 3,
&quot;followers&quot;: 10
}) ➞ [(&quot;dislikes&quot;, 3), (&quot;followers&quot;, 10), (&quot;likes&quot;, 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([&quot;p&quot;, &quot;s&quot;]) ➞ { &quot;p&quot;: &quot;P&quot;, &quot;s&quot;: &quot;S&quot; }

mapping([&quot;a&quot;, &quot;b&quot;, &quot;c&quot;]) ➞ { &quot;a&quot;: &quot;A&quot;, &quot;b&quot;: &quot;B&quot;, &quot;c&quot;: &quot;C&quot; }
mapping([&quot;a&quot;, &quot;v&quot;, &quot;y&quot;, &quot;z&quot;]) ➞ { &quot;a&quot;: &quot;A&quot;, &quot;v&quot;: &quot;V&quot;, &quot;y&quot;: &quot;Y&quot;, &quot;z&quot;: &quot;Z&quot; }
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(&quot;apples and bananas&quot;, &quot;u&quot;) ➞ &quot;upplus und bununus&quot;
vow_replace(&quot;cheese casserole&quot;, &quot;o&quot;) ➞ &quot;chooso cossorolo&quot;

vow_replace(&quot;stuffed jalapeno poppers&quot;, &quot;e&quot;) ➞ &quot;steffed jelepene peppers&quot;
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(&quot;to be or not to be!&quot;) ➞ &quot;To Be oR NoT To Be!&quot;
ascii_capitalize(&quot;THE LITTLE MERMAID&quot;) ➞ &quot;THe LiTTLe meRmaiD&quot;
ascii_capitalize(&quot;Oh what a beautiful morning.&quot;) ➞ &quot;oH wHaT a BeauTiFuL moRNiNg.&quot;

Answer1:

Here's the implementation of the `equal` function that takes three integers as input arguments and returns the number of integers which are of equal value.

python
```
def equal(a, b, c):
    if a == b == c:
        return 3
    elif a == b or b == c or a == c:
        return 2
    else:
        return 0
```

Here's how the function works:

- If all three integers are equal, the function returns 3.
- If any two of the integers are equal, the function returns 2.
- If all three integers are different, the function returns 0.

Examples:

python
```
print(equal(3, 4, 3)) # Output: 2
print(equal(1, 1, 1)) # Output: 3
print(equal(3, 4, 1)) # Output: 0
```

Answer2:

Here's the implementation of the `dict_to_list` function that takes a dictionary as input and returns a list of tuples containing the key-value pairs of the dictionary in alphabetical order.

python
```
def dict_to_list(d):
    return sorted(d.items())
```

Here's how the function works:

- The `items()` method of the dictionary is used to get a list of the key-value pairs of the dictionary.
- The `sorted()` function is used to sort the list of tuples in alphabetical order based on the keys.

Examples:

python
```
print(dict_to_list({"D": 1, "B": 2, "C": 3})) # Output: [('B', 2),
('C', 3), ('D', 1)]
print(dict_to_list({"likes": 2, "dislikes": 3, "followers": 10})) #
Output: [('dislikes', 3), ('followers', 10), ('likes', 2)]
```

Answer3:

Here's the implementation of the `mapping` function that takes a list of lowercase letters as input and returns a dictionary with each (key, value) pair being the (lowercase, uppercase) versions of the letter, respectively.

python
```
def mapping(lst):
    return {c: c.upper() for c in lst}
```

Here's how the function works:

- The input list `lst` is iterated over, and a dictionary comprehension is used to create a dictionary where each key is a lowercase letter in the input list and the corresponding value is the uppercase version of that letter.

Examples:

python
```
print(mapping(["p", "s"])) # Output: {'p': 'P', 's': 'S'}
print(mapping(["a", "b", "c"])) # Output: {'a': 'A', 'b': 'B', 'c':
'C'}
print(mapping(["a", "v", "y", "z"])) # Output: {'a': 'A', 'v': 'V',
'y': 'Y', 'z': 'Z'}
```

Answer4:

Here's the implementation of the `vow_replace` function that takes a string and a vowel as input and replaces all vowels in the string with the specified vowel.

python
```
def vow_replace(word, vowel):
    vowels = "aeiou"
    new_word = ""
    for letter in word:
        if letter in vowels:
            new_word += vowel
        else:
            new_word += letter
    return new_word
```

Here's how the function works:

- The input string `word` is iterated over, and each letter is checked if it is a vowel.
- If the letter is a vowel, it is replaced with the specified `vowel`.
- If the

5.Here's a Python function that takes a string as input, iterates over each character, and capitalizes a letter if its ASCII code is even, and returns its lower case version if its ASCII code is odd:

python
```
def ascii_capitalize(string):
    result = ""
    for char in string:
        ascii_code = ord(char)
        if ascii_code % 2 == 0:
            result += char.upper()
        else:
            result += char.lower()
    return result
```

Here's how the function works:

- We initialize an empty string called `result`.
- We iterate over each character in the input `string`.
- For each character, we get its ASCII code using the built-in `ord()` function.
- If the ASCII code is even (i.e., divisible by 2 with no remainder), we add its upper case version to the `result` string using the `upper()` method.
- Otherwise, we add its lower case version to the `result` string using the `lower()` method.
- Once we've processed all the characters in the input `string`, we return the `result` string.

Here are some examples of how to use the function:

python
```
>>> 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.'
```