Programming_Assingment25

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.

In [1]:

```
def equal(a,b,c):
    num = 0
    if a == b and a == c :
        num = 3
    elif a == b or a == c :
        num = 2
    else:
        num = 0
    return num
equal(3, 4, 3)
```

Out[1]:

```
2
```

In [2]:

```
equal(3, 4, 3)
```

Out[2]:

```
2
```

In [3]:

```
equal(1, 1, 1)
```

Out[3]:

```
3
```

In [4]:

```
equal(3, 4, 1)
```

Out[4]:

```
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.

In [5]:

```
def dict_to_list(d):
    return list(d.items())
```

In [6]:

```
dict_to_list({
    'D': 1,
    'B': 2,
    'C': 3
    })
```

Out[6]:

```
[('D', 1), ('B', 2), ('C', 3)]
```

In [7]:

```
dict_to_list({
    'likes': 2,
    'dislikes': 3,
    'followers': 10
    })
```

Out[7]:

```
[('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.

In [8]:

```
def mapping(lst):
    return {v.lower():v.upper() for v in lst}
```

In [9]:

```
mapping(['p', 's'])
```

Out[9]:

```
{'p': 'P', 's': 'S'}
```

In [10]:

```
mapping(['a', 'b', 'c'])
```

Out[10]:

```
{'a': 'A', 'b': 'B', 'c': 'C'}
```

In [11]:

```
mapping(['a', 'v', 'y', 'z'])
```

Out[11]:

```
{'a': 'A', 'v': 'V', 'y': 'Y', 'z': 'Z'}
```

In [12]:

```
mapping(['A', 'v', 'Y', 'z'])
```

Out[12]:

```
{'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.

In [13]:

```
def vow_replace(s,ch):
    vowel ='AEIOUaeiuo'
    s1 = []
    for i in range(len(s)):
        if s[i] in vowel:
            s1.append(ch)
        else:
            s1.append(s[i])

    return ''.join((s1))
```

In [14]:

```
vow_replace('apples and bananas', 'u')
```

Out[14]:

```
'upplus und bununus'
```

In [15]:

```
vow_replace('cheese casserole', 'o')
```

Out[15]:

```
'chooso cossorolo'
```

In [16]:

```
vow_replace('stuffed jalapeno poppers', 'e')
```

Out[16]:

```
'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.'

In [17]:
```
def ascii_capitalize(s):
    s1 = []
    for i in range(len(s)):
        if ord(s[i]) % 2 == 0:
            s1.append(s[i].upper())
        else:
            s1.append(s[i].lower())

    return "".join((s1))
```

In [18]:
```
ascii_capitalize('to be or not to be!')
```

Out[18]:
```
'To Be oR NoT To Be!'
```

In [19]:
```
ascii_capitalize('THE LITTLE MERMAID')
```

Out[19]:
```
'THe LiTTLe meRmaiD'
```

In [20]:
```
ascii_capitalize('Oh what a beautiful morning.')
```

Out[20]:
```
'oH wHaT a BeauTiFuL moRNiNg.'
```