**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.**

**ANS:-**

**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)**

**2**

**equal(3, 4, 3)**

**2**

**equal(1, 1, 1)**

**3**

**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.**

**ANS:-**

**def dict\_to\_list(d):**

**return list(d.items())**

**dict\_to\_list({**

**'D': 1,**

**'B': 2,**

**'C': 3**

**})**

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

**dict\_to\_list({**

**'likes': 2,**

**'dislikes': 3,**

**'followers': 10**

**})**

**[('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.**

**ANS:-**

**def mapping(lst):**

**return {v.lower():v.upper() for v in lst}**

**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'}**

**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") ➞ "choosocossorolo"**

**vow\_replace("stuffed jalapeno poppers", "e") ➞ "steffedjelepene peppers"**

### Notes

**All words will be lowercase. Y is not considered a vowel.**

**ANS:-**

**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))**

**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 oRNoT To Be!"**

**ascii\_capitalize("THE LITTLE MERMAID") ➞ "THeLiTTLemeRmaiD"**

**ascii\_capitalize("Oh what a beautiful morning.") ➞ "oHwHaT a BeauTiFuLmoRNiNg."**

**ANS:-**

**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))**

**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.'**