Assignment 4

Q.no- 1 Write the function countA(word) that takes in a word as argument and returns the number of 'a' in that word.

Examples

```
>>> countA("apple")
1
>>> countA("Apple")
0
>>> countA("Banana")
3
```

Ans - def countA(word):
    count=0
    for x in word:
        if x == 'a':
            count=count+1
    return count

Q.no- 2 Write the function countLetter(word, letter) that takes in a word and a letter as arguments and returns the number of occurrence of that letter in the word.

Examples

```
>>> countLetter("apple", "p")
2
>>> countLetter("Apple", "a")
0
>>> countLetter("Banana", "n")
2
```

Ans - def countLetter(word, letter):
    count=0
    for x in word:
        if x == letter:
            count=count+1
    return count

Q.no- 3 Write a function removeLetter(word, letter) that takes in a word and a letter as arguments and remove all the occurrence of that particular letter from the word. The function will returns the remaining leters in the word.

Examples

```
>>> removeLetter("apple", "p")
'ale'
>>> removeLetter("microsoft", "o")
'micrsft'
>>> removeLetter("google", "g")
'oole'
```

Ans -
```
def removeLetter(word, letter):
    str=''
    for x in word:
        if x != letter:
            str=str+x
    return str
```

Q.no- 4 Write the function changeCase(word) that changes the case of all the letters in a word and returns the new word.

Examples

```
>>> changeCase('aPPle')
"AppLE"
>>> changeCase('BaNaNa')
'bAnAnA'
```

Ans -
```
def changeCase(word):
    return word.swapcase()
```

Q.no- 5 Write the function search(word, substring) that takes in a word and a substring as arguments and returns the position (0 indexed) of the substring if it is found in the word. The function returns -1 if the substring is not found.

Examples

```
>>> search("apple", 'p')
1
>>> search("google", 'apple')
-1
>>> search("google", 'p')
-1
>>> search("google", 'oo')
1
```

Ans - def search(word, substring):
    return word.find(substring)

Q.no- 6 A string contains a sequence of characters. Elements within a string can be accessed using index that starts from 0. Write the function getChar(word, pos) that takes in a word and a number as argument and returns the character at that position.

Examples

```
>>> s = "Hello"
 >>> s[0]
'H'
>>> s[-1]
'o'
>>> getChar("apple", 2)
p
>>> getChar("google", 0)
g
>>> getChar("google", 10)
Invalid Range.
```

Ans - def getChar(word, pos):
    for x in word:
        if pos>len(word):
            return 'Invalid Range.'
        else:
            return word[pos]

Q.no- 7 Write a function countVowels(word) that takes in a word as an argument and returns the number of vowels ('a', 'e', 'i', 'o', 'u') in the word.

Examples

```
>>> countVowels('apple')
 2
>>> countVowels('microsoft')
3
>>> countVowels('google')
3
```

Ans - def countVowels(word):
    count=0
    for x in word:
        if x in 'aeiou':
            count=count+1
    return count

Q.no- 8 Write the function getVowels(word) that takes in a word as an argument and returns the vowels ('a', 'e', 'i', 'o', 'u') in that word.

Examples

```
>>> getVowels("apple")
['a', 'e']
>>> getVowels("Apple")
['A', 'e']
>>> getVowels("Banana")
['a', 'a', 'a']
```

Ans -
```
def getVowels(word):
    vowels='aeiouAEIOU'
    str=[]
    for x in word:
        if x in vowels:
            str.append(x)
    return str
```

Q.no- 9 Write the function capitalizeVowels(word) that returns the word with all the vowels capitalized.

Examples

```
>>> capitalizeVowels('apple')
'ApplE'
>>> capitalizeVowels('google')
'gOOglE'
```

Ans -
```
def capitalizeVowels(word):
    for x in word:
        if x=='a' or x=='e' or x=='i' or x=='o' or x=='u':
            a=x.upper()
            word= word.replace(x,a)
    return word
```

Q.no- 10 Write the function startEndVowels(word) that returns True if the word starts and ends with vowels.

Examples

```
>>> startEndVowels('apple')
True
>>> startEndVowels('google')
False
>>> startEndVowels('A')
```

```
    True
    >>> startEndVowels('')
    False
```

Ans - def startEndVowels(word):
    vowels='aeiouAEIOU'
    return len(word) > 0 and word[0] in vowels and word[-1] in vowels

Q.no- 11 Write the function removeVowels(word) that removes all the vowels ('a', 'e', 'i', 'o', 'u') in a word and returns the remaining letters in the word.

Examples

```
   >>> removeVowels('apple')
    "ppl"
    >>> removeVowels('Apple')
    "ppl"
    >>> removeVowels('Banana')
    'Bnn'
```

Ans - def removeVowels(word):
    str=''
    for x in word:
        if x not in 'aeiouAEIOU':
            str=str+x
    return str

Q.no- 12 Write the function reverseWord(word) that returns the word in the reverse order.

Examples

```
   >>> reverseWord('apple')
    'elppa'
    >>> reverseWord('google')
    'elgoog'
```

Ans - def reverseWord(word):
    return word[-1::-1]

Q.no- 13 Write the function isReverse(word1, word2) that takes two words as arguments and returns True is the second word is the reverse of the first word.

Examples

```
   >>> isReverse('apple', 'elppa')
    True
```

```
>>> isReverse('google', 'apple')
False
>>> isReverse('google', 'elgoog')
True
>>> isReverse('apple', 'alppe')
False
```

Ans - def isReverse(word1, word2):
    if word1==word2[-1::-1]:
        return True
    else:
        return False

Q.no- 14 Write the function startWithVowel(word) that takes in a word as argument and returns a substring that starts with the first vowel found in the word. The function returns 'No vowel' if the word does not contain vowel.

Examples

```
>>> startWithVowel('apple')
'apple'
>>> startWithVowel('google')
'oogle'
>>> startWithVowel('xyz')
'No vowel'
```

Ans - def startWithVowel(word):
    if word[0] in 'aeiou':
        return word
    else:
        for i in range(1, len(word)):
            if word[i] in 'aeiou': return word[i:]
        return "No vowel"

Q.no- 15 Write the function getCommonLetters(word1, word2) that takes in two words as arguments and returns a new string that contains letters found in both string. Ignore repeated letters and sort the result in alphabetical order.

Examples

```
>>> getCommonLetters('apple','google')
'el'
>>> getCommonLetters('microsoft','apple')
''
```

```
>>> getCommonLetters('microsoft','google')
'o'
```

Ans - def getCommonLetters(word1, word2):
return ''.join(sorted(set(word1) & set(word2)))

Q.no- 16 Write a function mirrorText(word1, word2) that takes in 2 words as arguments and returns a new word in the following order: word1word2word2word1.

Examples

```
>>> mirrorText('hello','world')
'helloworldworldhello'
>>> mirrorText('apple','orange')
'appleorangeorangeapple'
>>> mirrorText('google','yahoo')
'googleyahooyahoogoogle'
```

Ans - def mirrorText(word1,word2):
return word1+word2+word2+word1

Q.no- 17 Write a function echoWord(word) that takes in a word as arguments and returns a word that repeats itself based on the number of letter in the word.

Examples

```
>>> echoWord('hi')
'hihi'
>>> echoWord('apple')
'appleappleappleappleapple'
>>> echoWord('ice')
'iceiceice'
```

Ans - def echoWord(word):
return word*len(word)

Q.no- 18 Write a function rightJustify(word) that takes in a word as argument and return a word with leading spaces so that the last letter of the word is in column 50 of the display.

Examples

```
>>> rightJustify('apple')
'                                              apple'
>>> rightJustify('google')
'                                             google'
>>> rightJustify('microsoft')
'                                          microsoft'
```

Ans - def rightJustify(word):
    return word.rjust(50)

Q.no- 19 A palindrome is a word, phrase, number or other sequence of units that can be read the same way in either direction. Write a function that determines whether the given word or number is a palindrome.

Examples

```
>>> isPalindrome("Racecar")
True
>>> isPalindrome(121)
True
>>> isPalindrome("Never")
False
>>> isPalindrome("level")
True
>>> isPalindrome("")
False
```

Ans - def isPalindrome(word):
    word=str(word)
    reverse=word[::-1]
    if len(word)!=0:
        if word==reverse:
            return True
    return False

Q.no- 20 Write a function isInAlphabeticalOrder(word) that takes in a word as argument and returns True if the word contains letters that are arranged in alphabetical order. For example, the letter 'c' should not appear before the letter 'a'.

Examples

```
>>> isInAlphabeticalOrder('app')
True
>>> isInAlphabeticalOrder('apple')
False
>>> isInAlphabeticalOrder('goo')
True
>>> isInAlphabeticalOrder('google')
False
```

Ans - def isInAlphabeticalOrder(word):
    for i in range(len(word) - 1):
        if word[i] > word[i + 1]:
            return False
    return True

Q.no- 21 Write a function isAllLettersUsed(word, required) that takes in a **word** as the first argument and returns True if the word contains all the letters found in the second argument.

Examples

```
>>> isAllLettersUsed('apple', 'apple')
True
>>> isAllLettersUsed('apple', 'google')
False
>>> isAllLettersUsed('learning python', 'google')
True
>>> isAllLettersUsed('learning python', 'apple')
True
```

Ans - def isAllLettersUsed(word, required):
    for x in required:
        if x in word:
            return True
        return False

Q.no- 22 Write a function isTripleDouble(word) that takes in a word as argument and returns True if the word contains three consecutive double letters.

Examples

```
>>> isTripleDouble('appllee')
True
>>> isTripleDouble('aapplee')
False
>>> isTripleDouble('applle')
False
```

Ans - def isTripleDouble(word):
    i = 0
    count = 0
    while i < len(word)-1:
        if word[i] == word[i+1]:

```
                    count = count + 1
                    if count == 3:
                            return True
                    i = i + 2
            else:
                    count = 0
                    i = i + 1
    return False
```

Q.no- 23 Write a function splitWord(word, numOfChar) that takes in a word and a number as arguments. The function will split the word into smaller segments with each segment containing the number of letter specified in the numOfChar argument. These segments are stored and returned in a list.

Examples

```
>>> splitWord('google', 2)
['go', 'og', 'le']
>>> splitWord('google', 3)
['goo', 'gle']
>>> splitWord('apple', 1)
['a', 'p', 'p', 'l', 'e']
>>> splitWord('apple', 4)
['appl', 'e']
```

Ans -
```
def splitWord(word, numOfChar):
    lst=[]
    for x in range(0, len(word), numOfChar):
        lst.append(word[x:x+numOfChar])
    return lst
```

Q.no- 24 An anagram is a word formed by reordering the letters of another word. Write a function isAnagram(w1, w2) that takes in two words as arguments and return True if one word is an anagram of the other word.

Examples

```
>>> isAnagram('google', 'gogole')
True
>>> isAnagram('google', 'gogoll')
False
>>> isAnagram('google', 'gogogo')
False
>>> isAnagram('Google', 'google')
True
```

Ans - def isAnagram(w1, w2):
```
    s1=w1.lower()
    s2=w2.lower()
    if (sorted(s1)==sorted(s2)):
        return True
    return False
```