**SESSION 2 (LINK ONE**)

A list behaves like a container, and it is able to contain more than one value. Create a list with the values as shown in the example below

aList = ['Hello',0,20.0,'World']

A list can be modified, and more elements can be added to an existing list. Use the append(x) function to add some more items to the list to produce the same content shown in the sample below.

aList = ['Hello', 0]

aList.append(20.0)

aList.append("World")

A list can be modified, and elements can be removed from an existing list. Use the remove(x) function to remove some items from the list shown in the sample given below so that the list is left with the following content: ['hello', 'python','programming'].

aList = ['hello', 'i', 'love', 'python', 'programming']

aList.pop(1)

aList.pop(1)

Write a function addNumbersInList(numbers) to add all the numbers in a list. To access each element in a list, you can use the statement '**for** num **in** numbers:'.

def addNumbersInList(numbers):

total=0

for element in numbers:

total+=element

return total

Write a function addOddNumbers(numbers) to add all the odd numbers in a list. To access each element in a list, you can use the statement 'for num in numbers:'.

def addOddNumbers(numbers):

total=0

for element in numbers:

if element%2==1:

total+=element

return total

Write a function countOddNumbers(numbers) to count the number of odd numbers in a list.

def countOddNumbers(numbers):

alist=[]

for element in numbers:

if element%2==1:

alist.append(element)

return len(alist)

Write a function getEvenNumbers(numbers) to return all the even numbers in a list.

def getEvenNumbers(numbers):

alist=[]

for element in numbers:

if element%2==0:

alist.append(element)

return alist

Write a function removeFirstAndLast(list) that takes in a list as an argument and remove the first and last elements from the list. The function will return a list with the remaining items.

def removeFirstAndLast(numbers):

if len(numbers)==0:

print("the list is empty")

else:

numbers.pop(0)

if len(numbers)>0:

numbers.pop()

return numbers

Write a function getMaxNumber(numbers) that returns the maximum number in a list.

def getMaxNumber(numbers):

if len(numbers)==0:

return ('N.A')

else:

return max(numbers)

Write a function getMinNumber(numbers) that returns the minimum number in a list.

def getMinNumber(numbers):

if len(numbers)==0:

return ('N.A')

else:

return min(numbers)

Write a function that does matrix multiplication.  
   The product of a **m**x**n** matrix with a **n**x**p** matrix results in a **m**x**p** matrix.  
   A mxn matrix, with m rows and n columns, can be represented using nested lists.  
   Am,n = [ [x11, x12, ..., x1n], ..., [xm1, ..., xmn] ]

def MatrixProduct(a, b):  
  
c = [[0 for i in range(len(b[0]))] for j in range(len(a))]  
  
for i in range(len(a)):  
for j in range(len(b[0])):  
for k in range(len(b)):  
c[i][j] += a[i][k] \* b[k][j]  
return c

A mxn matrix, m rows and n columns, can be represented using nested lists. Write a function that returns the diminensions of a matrix.

ef matrixDimensions(m):

for i in range(len(m)-1):

if len(m[i])!=len(m[i+1]):

return ("This is not a valid matrix.")

for i in range(len(m)-1):

if len(m[i])==len(m[i+1]):

return ("This is a {0}x{1} matrix.".format(len(m),len(m[0])))

if len(m)==1:

return ("This is a {}x{} matrix.".format(len(m),len(m[0])))

Write a function combine(la, lb) that takes in two lists and return a list with the contents of both list sorted in ascending order.

def combine(la, lb):

ma=la+lb

ma.sort()

return ma

The *transpose* of a matrix M, denoted MT, is formed by interchanging the rows and columns of M. That is, a mxn matrix is transformed into a nxm matrix. [MT]ij = [M]ji. Write a function that returns the transpose of a matrix.

def transpose(matrix):  
  
   trans = []  
   pose = []  
   for j in range(len(matrix[0])):  
       for i in range(len(matrix)):  
           trans.append(matrix[i][j])  
       pose.append(trans)  
       trans = []  
  
   return pose

Write a function calCumulativeSum(numbers) that takes in a list of numbers as argument and returns the cumulative sum of the list. That is, the new list where the *i* element is the sum of the first *i* + 1 elements from the original list. For example, the cumulative sum of [1, 2, 3] is [1, 3, 6].

def calCumulativeSum(numbers):

newlist=[]

total=0

for num in numbers:

total+=num

newlist.append(total)

return newlist

Write a function combineList(list1, list2) that takes in two lists as arguments and return a list that combines all the elements in the two list.

def combineList(list1, list2):

x=list1+list2

return x

Write a function (list1, list2) that takes in two lists as arguments and return a list that is the result of removing elements from list1 that can be found in list2.

def subtractList(list1, list2):

newlist=[]

for i in list1:

if i not in list2:

newlist.append(i)

return newlist

Write a function getNumbers(number) that takes in a number as argument and return a list of numbers as shown in the samples given below.

def getNumbers(num):

list1=[]

for i in range(-num,num+1,2):

list1.append(i\*i)

return list1

Write a function getSumOfFirstDigit(numList) that takes in a list of positive numbers and returns the sum of all the first digit in the list.

def getSumOfFirstDigit(num):

total=0

for number in num:

temp=str(number)

total+=int(temp[0])

return total

**SESSION 2(LINK 2)**

Write the function countA(word) that takes in a word as argument and returns the number of 'a' in that word.

def countA(word):

x=word.count('a')

return x

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.

def countLetter(word, letter):

x=word.count(letter)

return x

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.

def removeLetter(word, letter):

x=word.replace(letter,'')

return x

Write the function changeCase(word) that changes the case of all the letters in a word and returns the new word.

def changeCase(word):

x=word.swapcase()

return x

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.

def getChar(word, pos):

if len(word)>pos:

m=word[pos]

return (m)

else:

return ("Invalid Range.")

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.

def countVowels(word):

total=0

for i in word:

if i in 'aeiou':

total+=1

return total

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.

def getVowels(word):

newlist=[]

for val in word:

if val in 'aeiouAEIOU':

newlist.append(val)

return newlist

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.

def getChar(word, pos):

if len(word)>pos:

m=word[pos]

return (m)

else:

return ("Invalid Range.")

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.

def countVowels(word):

total=0

for i in word:

if i in 'aeiou':

total+=1

return total

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.

def getVowels(word):

newlist=[]

for val in word:

if val in 'aeiouAEIOU':

newlist.append(val)

return newlist

Write the function capitalizeVowels(word) that returns the word with all the vowels capitalized.

def capitalizeVowels(word):

wordlist=[]

for i in word:

if i in 'aeiou':

wordlist.append(i.upper())

else:

wordlist.append(i)

return (''.join(wordlist))

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.

def removeVowels(word):

wordlist=[]

for i in word:

if (i in "aeiou") or(i in "AEIOU"):

continue

else:

wordlist.append(i)

return ''.join(wordlist)

Write the function reverseWord(word) that returns the word in the reverse order.

def reverseWord(word):

return word[::-1]

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.

def isReverse(word1, word2):

if word1==word2[::-1]:

return True

else:

return False

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.

def startWithVowel(word):

wordlist=[]

index=0

for i in word:

if i in "aeiou":

m=word[index:]

return m

index+=1

for i in word:

if i not in "aeiou":

return "No vowel"

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.

def getCommonLetters(word1, word2):

newlist=[]

for i in word1:

if i in word2 :

if i in newlist:

continue

else:

newlist.append(i)

x=newlist.sort()

return ''.join(newlist)

Write a function mirrorText(word1, word2) that takes in 2 words as arguments and returns a new word in the following order: word1word2word2word1.

def mirrorText(word1,word2):

x=word1+word2+word2+word1

return x

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.

def echoWord(word):

x=word\*len(word)

return x

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.

def isPalindrome(word):

word=str(word)

if word=='':

return False

if word==word[::-1]:

return True

else:

return False

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'.

def isInAlphabeticalOrder(word):

newlist=[]

for i in word:

newlist.append(i)

x=newlist.sort()

y=''.join(newlist)

if y==word:

return True

else:

return False

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.

def isAllLettersUsed(word, required):

for i in word:

if i in required:

return True

else:

return False

Write a function isTripleDouble(word) that takes in a word as argument and returns True if the word contains three consecutive double letters.

def isTripleDouble(word):

i=0

counts=0

while i<len(word)-1:

if word[i]==word[i+1]:

counts+=1

if counts==3:

return True

i+=2

else:

counts=0

i+=1

return False

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.

def splitWord(word, length):

r = []

for count in range(0, len(word), length):

r.append(word[count:count+length])

return r

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.

def isAnagram(w1, w2):

for i in w1.lower():

if i in w2.lower():

continue

else:

return False

return True

**SESSION 2(LINK 3)**

The dictionary data structure consists of key-value data pair.

# Initialize dictionary "contactinfo" with the values

# as shown in above examples. Hint: The key is a string

# literal while the value is a dictionary type.

contactinfo={"Tom":{'Email':'tom@gmail.com','Phone':61234567},"Sally":{'Email':'sally@hotmail.com','Phone':67654321}}

contactinfo

In gene expression, mRNA is transcribed from a DNA template. The 4 nucleotide bases of A, T, C, G corresponds to the U, A, G, C bases of the mRNA. Write a function that returns the mRNA transcript given the sequence of a DNA strand.

# Use a dictionary to provide the mapping of DNA to RNA bases.

def mRNAtranscription(dna\_template):

dna2rna = {'A':'U','T':'A','C':'G','G':'C' }

mRNA = ''

for base in dna\_template:

x=dna2rna[base]

mRNA=mRNA+x

return mRNA

A DNA strand consisting of the 4 nucleotide bases is usually represented with a string of letters: A,T, C, G. Write a function that computes the base composition of a given DNA sequence.

def baseComposition(dna\_seq):

DNA={'A':0,'C':0,'T':0,'G':0}

for base in dna\_seq:

if base in DNA:

DNA[base]+=1

return DNA

Write a function countLetters(word) that takes in a word as argument and returns a dictionary that counts the number of times each letter appears.

def countLetters(word):

countword={}

for letter in word:

if letter in countword:

countword[letter]+=1

else:

countword[letter]=1

return countword

The determinant of a 2x2 matrix is the product of the elements on the main diagonal minus the product of the elements off the main diagonal.

def det(M):

x=(M[0][0]\*M[1][1])-(M[0][1]\*M[1][0])

return x

Write a function hasSameContent(t1, t2) that takes in two tuples as arguments and return True if both tuples contain the same items.

def hasSameContent(t1, t2):

if len(t1)!=len(t2):

return False

for element in t1:

if element in t2:

continue

else:

return False

return True

Write a function sumNumbers(\*args) that takes in a variable-length argument list of numbers and returns the sum of the numbers.

def sumNumbers(\*args):

sums=0

for element in args:

sums+=element

return sums

Write a function commonElements(t1, t2) that takes in 2 tuples as arguments and returns a sorted tuple containing elements that are found in both tuples.

def commonElements(t1,t2):

t3=[]

for element in t1:

if element in t2:

t3.append(element)

return tuple(t3)

Write a function sortByIndex(aList) that takes in a list of tuple in the following format: (index, value) and returns a new tuple with its elements sorted based on the index.

def sortByIndex(aList):

t = tuple()

for i in range(len(aList)):

t += (sorted(aList)[i][1],)

return t

Write a function sortByLength(t, order) that takes in a tuple of string and returns a new tuple with its elements sorted by the length of the string. The order of sorting is based on the value of the second argument: 'asc' or 'des'.

def sortByLength(t, order):

tup = ()

for i in t:

tup += ((len(i), i),)

if order == 'asc':

tup = sorted(tup)

elif order == 'des':

tup = sorted(tup, reverse = True)

ans = ()

for i in range(len(t)):

ans += (tup[i][1],)

return ans