Topic 6: Question 1

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.

**Examples**

>>> aList[0]

'Hello'

>>> aList[1:3]

[0, 20.0]

>>> aList[3]

'World'

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

Topic 6: Question 2

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.

**Examples**

>>> aList[0]

'Hello'

>>> aList[1:3]

[0, 20.0]

>>> aList[3]

'World'

aList = ['Hello', 0]

aList.append(20.0)

aList.append('World')

Topic 6: Question 3

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.remove('i')

aList.remove('love')

Topic 6: Question 4

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:'.

**Examples**

>>> addNumbersInList([])

0

>>> addNumbersInList([10, 20, 30])

60

>>> addNumbersInList([-10, -20, 30])

def addNumbersInList(numbers):

return sum(i for i in numbers)

Topic 6: Question 5

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:'.

**Examples**

>>> addOddNumbers([1, 4, 8, 9])

10

>>> addOddNumbers(range(1, 20, 3))

40

>>> addOddNumbers([])

0

def addOddNumbers(numbers):

return sum(i for i in numbers if i%2 ==1)

Topic 6: Question 6

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

def countOddNumbers(numbers):

s=0

for num in numbers:

if num%2==1:

s+=1

return s

Topic 6: Question 7

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

def getEvenNumbers(numbers):

s=[]

for num in numbers:

if num%2==0:

s.append(num)

return s

Topic 6: Question 8

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.

def removeFirstAndLast(numbers):

Topic 6: Question 9

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

def getMaxNumber(numbers):

if(numbers == []):

return "N.A"

a = list(numbers)

return max(a)

Topic 6: Question 10

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

def getMinNumber(numbers):

if(numbers == []):

return "N.A"

s=list(numbers)

return min(s)

Topic 6: Question 11

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):

Topic 6: Question 16

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):

c=(la+lb)

return sorted(c)

Topic 6: Question 17

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):

return map(list, zip(\*matrix))

Topic 6: Question 20

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):

return (list1+list2)

Question 21

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):

new\_list = [num for num in list1 if num not in list2]

return new\_list

Topic 6: Question 22

Write a function countLetters(word) that takes in a word as argument and returns a list of tuples that shows the number of times each letter appears. The letters must be sorted in alphabetical order.

def countLetters(word)

Topic 6: Question 26

List comprehension offers a concise way to derive a new list from an existing list or sequence. Given a list of numbers, write a function that returns the numbers that are greater than the average

def getAboveAverage(nums):

average=sum(nums)/len(nums)

return [ x for x in nums if x >average ] # complete the list comprehension.

Topic 7: Question 1

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

def countA(word):

return word.count('a')

Topic 7: Question 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.

def countLetter(word, letter):

return word.count(letter)

Topic 7: Question 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.

def removeLetter(word, letter):

return word.replace(letter,'')

Topic 7: Question 4

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

def changeCase(word):

result = ""

for letter in word:

if letter == letter.upper():

result += letter.lower()

else:

result += letter.upper()

return result

Topic 7: Question 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.

def search(word, substring):

if substring in word:

return len(substring)

return -1

Topic 7: Question 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.

def getChar(word, pos):

Topic 7: Question 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.

def countVowels(word):

counter=0

for i in word:

if i in ('a', 'e', 'i', 'o', 'u'):

counter+=1

return(counter)

Topic 7: Question 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.

def getVowels(word):

array=[]

for i in word:

if i in ('a','e','i','o','u','A','E','I','O','U'):

array.append(i)

return(array)

Topic 7: Question 9

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

def capitalizeVowels

Topic 7: Question 10

Write the function startEndVowels(word) that returns True if the word starts and ends with vowels.

def startEndVowels(word):

vowels = tuple("aeiouAEIOU")

return word.startswith(vowels) and word.endswith(vowels)

Topic 7: Question 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.

def removeVowels(word):

for i in "aeiouAEIOU":

word = word.replace(i,"")

return word

Topic 7: Question 12

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

def reverseWord

Topic 7: Question 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.

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"

Topic 7: Question 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.

def getCommonLetters(word1, word2):

Topic 7: Question 16

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):

return (word1+word2+word2+word1)

Topic 7: Question 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.

def echoWord(word):

for i in word:

return (len(word)\*word)

opic 7: Question 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'.

def isInAlphabeticalOrder(word):

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

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

return False

return True

Topic 7: Question 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.

def isAllLettersUsed(word, required):

for i in word:

if i in required:

return True

else:

return False

opic 8: Question 1

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

**Examples**

>>> a = {} *# empty dictionary*

>>>type(a)

< type 'dict' >

>>> book = {"Author":"Lewis Carroll"}

>>> book["Title"] = "Alice's Adventures in Wonderland"

>>> book

{'Title': "Alice's Adventures in Wonderland", 'Author': 'Lewis Carroll'}

>>> contactinfo["Tom"]

{'Email':'tom@gmail.com', 'Phone':61234567}

>>> contactinfo["Sally"]

{'Email':'sally@hotmail.com', 'Phone':67654321}

# 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 ={}

contactinfo["Tom"]={'Email':'tom@gmail.com', 'Phone':61234567}

contactinfo["Sally"]={'Email':'sally@hotmail.com', 'Phone':67654321}

Topic 8: Question 5

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:

mRNA+=dna2rna.get(base)

return mRNA

Topic 8: Question 6

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):

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

for letter in dna\_seq:

if letter in counts:

counts[letter]+=1

return counts

Topic 8: Question 7

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):

result = { }

letters = ""

for i in word:

if i not in letters:

letters += i

result[i] = 0

for i in word:

if i in letters:

result[i] += 1

return result

Topic 8: Question 8

Write a function reverseLookup(dictionary, value) that takes in a dictionary and a value as arguments and returns a sorted list of all keys that contains the value. The function will return an empty list if no match is found.

def reverseLookup(dictionary, value):

result=[]

for i in dictionary:

if dictionary[i]==value:

result.append(i)

return sorted(result)

Topic 9: Question 4

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

if sorted(t1)==sorted(t2):

return True

else:

return False

Topic 9: Question 5

In python, it is possible to pass in variable-length argument to a function by using the following notation: \*args. The argument is passed in as a tuple.

def funA(\*args):

print type(args)

print args

>>> funA(1, 3, 5)

<type 'tuple'>

(1, 3, 5)

>>> funA('a', 'b', 'c', 'd')

<type 'tuple'>

('a', 'b', 'c', 'd')

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):

return (sum(args)

Topic 9: Question 7

Write a function removeCommonElements(t1, t2) that takes in 2 tuples as arguments and returns a sorted tuple containing elements that are **not** found in both tuples.

def removeCommonElements(first,second):

return tuple((sorted(set(first) ^ set(second))))

Topic 9: Question 8

Write a function shiftByTwo(\*args) that takes in variable-length argument and returns a tuple with its elements shifted to the right by two indices. See samples given below.

def shiftByTwo(\*a):

return a[len(a)-2:len(a)] + a[0:len(a)-2]