

# 

[Lists](#_6372urs32nir) **14**

[Dictionaries](#_7jtgmhhs4eeq) **25**

[Tuples](#_91la6da7d3d6) **29**

[Classes & Objects](#_ujst7afm07rz) **31**

# 

# Lists

1. Write a Python function to sum all the items in a list.
2. Write a Python function to multiply all the items in a list.
3. Write a Python function to get the largest number from a list.
4. Write a Python function to get the smallest number from a list.
5. Write a Python function to count the number of strings where the string length is 2 or more and the first and last character are the same from a given list of strings.   
   Sample List : ['abc', 'xyz', 'aba', '1221']

Expected Result : 2

1. Write a Python function to remove duplicates from a list.
2. Write a Python function to check if a list is empty or not.
3. Write a Python function to clone or copy a list.
4. Write a Python function to find the list of words that are longer than n from a given list of words.
5. Write a Python function that takes two lists and returns True if they have at least one common member.
6. Write a Python function to print a specified list after removing the 0th, 4th and 5th elements.

Sample List : ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

Expected Output : ['Green', 'White', 'Black']

1. Write a Python function to print the numbers of a specified list after removing even numbers from it.
2. Write a Python function to get the difference between the two lists.
3. Write a Python function to convert a list of characters into a string.
4. Write a Python function to find the index of an item in a specified list.
5. Write a Python function to append a list to the second list.
6. Write a Python function to select an item randomly from a list.
7. Write a Python function to find the second smallest number in a list.
8. Write a Python function to find the second largest number in a list.
9. Write a Python function to get unique values from a list.
10. Write a Python function to count the number of elements in a list within a specified range.
11. Write a Python function to check whether a list contains a sublist.
12. Write a Python function to create a list by concatenating a given list which range goes from 1 to n.   
    Sample list : ['p', 'q']

n =5

Sample Output : ['p1', 'q1', 'p2', 'q2', 'p3', 'q3', 'p4', 'q4', 'p5', 'q5']

1. Write a Python function to change the position of every n-th value with the (n+1)th in a list.

Sample list: [0,1,2,3,4,5]

Expected Output: [1, 0, 3, 2, 5, 4]

1. Write a Python function to convert a list of multiple integers into a single integer.

Sample list: [11, 33, 50]

Expected Output: 113350

1. Write a Python function to split a list based on the first character of a word.
2. Write a Python function to insert an element before each element of a list.
3. Write a Python function to replace the last element in a list with another list.

Sample data : [1, 3, 5, 7, 9, 10], [2, 4, 6, 8]

Expected Output: [1, 3, 5, 7, 9, 2, 4, 6, 8]

1. Write a Python function to check whether the n-th element exists in a given list.
2. Write a Python function to insert a given string at the beginning of all items in a list.

Sample list : [1,2,3,4], string : emp

Expected output : ['emp1', 'emp2', 'emp3', 'emp4']

1. Write a Python function to move all zero digits to the end of a given list of numbers.

Expected output:

Original list:

[3, 4, 0, 0, 0, 6, 2, 0, 6, 7, 6, 0, 0, 0, 9, 10, 7, 4, 4, 5, 3, 0, 0, 2, 9, 7, 1]

Move all zero digits to end of the said list of numbers:

[3, 4, 6, 2, 6, 7, 6, 9, 10, 7, 4, 4, 5, 3, 2, 9, 7, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

1. Write a Python function to find the list in a list of lists whose sum of elements is the highest.

Sample lists: [1,2,3], [4,5,6], [10,11,12], [7,8,9]

Expected Output: [10, 11, 12]

1. Write a Python function to find all the values in a list that are greater than a specified number.
2. Write a Python function to remove consecutive duplicates of a given list.

Original list:

[0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 8, 9, 4, 4]

After removing consecutive duplicates:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 4]

1. Write a Python function to pack consecutive duplicates of a given list element into sublists.

Original list:

[0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 8, 9, 4, 4]

After packing consecutive duplicates of the said list elements into sublists:

[[0, 0], [1], [2], [3], [4, 4], [5], [6, 6, 6], [7], [8], [9], [4, 4]]

1. Write a Python function to split a given list into two parts where the length of the first part of the list is given.

Original list:

[1, 1, 2, 3, 4, 4, 5, 1]

Length of the first part of the list: 3

Splitted the said list into two parts:

([1, 1, 2], [3, 4, 4, 5, 1])

1. Write a Python function to remove the K'th element from a given list, return the new list.

Original list:

[1, 1, 2, 3, 4, 4, 5, 1]

After removing an element at the kth position of the said list:

[1, 1, 3, 4, 4, 5, 1]

1. Write a Python function to insert an element at a specified position into a given list.

Original list:

[1, 1, 2, 3, 4, 4, 5, 1]

After inserting an element at kth position in the said list:

[1, 1, 12, 2, 3, 4, 4, 5, 1]

1. Write a Python function to extract a given number of randomly selected elements from a given list.

Original list:

[1, 1, 2, 3, 4, 4, 5, 1]

umber = 3

[4, 4, 1]

1. Write a Python function to count the number of lists in a given list of lists.

Original list:

[[1, 3], [5, 7], [9, 11], [13, 15, 17]]

Number of lists in said list of lists:

4

Original list:

[[2, 4], [[6, 8], [4, 5, 8]], [10, 12, 14]]

Number of lists in said list of lists:

3

1. Write a Python function to sort a given matrix in ascending order according to the sum of its rows.

Original Matrix:

[[1, 2, 3], [2, 4, 5], [1, 1, 1]]

Sort the said matrix in ascending order according to the sum of its rows

[[1, 1, 1], [1, 2, 3], [2, 4, 5]]

Original Matrix:

[[1, 2, 3], [-2, 4, -5], [1, -1, 1]]

Sort the said matrix in ascending order according to the sum of its rows

[[-2, 4, -5], [1, -1, 1], [1, 2, 3]]

1. Write a Python function to extract specified size of strings from a given list of string values.

Original list:

['Python', 'list', 'exercises', 'practice', 'solution']

length of the string to extract:

8

After extracting strings of specified length from the said list:

['practice', 'solution']

1. Write a Python function to find the difference between consecutive numbers in a given list.

Original list:

[1, 1, 3, 4, 4, 5, 6, 7]

Difference between consecutive numbers of the said list:

[0, 2, 1, 0, 1, 1, 1]

Original list:

[4, 5, 8, 9, 6, 10]

Difference between consecutive numbers of the said list:

[1, 3, 1, -3, 4]

1. Write a Python function to compute the average of two given lists.

Original list:

[1, 1, 3, 4, 4, 5, 6, 7]

[0, 1, 2, 3, 4, 4, 5, 7, 8]

Average of two lists:

3.823529411764706

1. Write a Python function to count integers in a given mixed list.

Original list:

[1, 'abcd', 3, 1.2, 4, 'xyz', 5, 'pqr', 7, -5, -12.22]

Number of integers in the said mixed list:

6

1. Write a Python function to remove a specified column from a given nested list.

Original Nested list:

[[1, 2, 3], [2, 4, 5], [1, 1, 1]]

After removing 1st column:

[[2, 3], [4, 5], [1, 1]]

Original Nested list:

[[1, 2, 3], [-2, 4, -5], [1, -1, 1]]

After removing 3rd column:

[[1, 2], [-2, 4], [1, -1]]

1. Write a Python function to extract a specified column from a given nested list.

Original Nested list:

[[1, 2, 3], [2, 4, 5], [1, 1, 1]]

Extract 1st column:

[1, 2, 1]

Original Nested list:

[[1, 2, 3], [-2, 4, -5], [1, -1, 1]]

Extract 3rd column:

[3, -5, 1]

1. Write a Python function to find the item with maximum occurrences in a given list.

Original list:

[2, 3, 8, 4, 7, 9, 8, 2, 6, 5, 1, 6, 1, 2, 3, 4, 6, 9, 1, 2]

Item with maximum occurrences of the said list:

2

1. Write a Python function to check whether a specified list is sorted or not.

Original list:

[1, 2, 4, 6, 8, 10, 12, 14, 16, 17]

The said list is sorted?

True

Original list:

[1, 2, 4, 6, 8, 10, 12, 14, 16, 17]

The said list is sorted?

False

1. Write a Python function to remove all elements from a given list present in another list.

Original lists:

list1: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

list2: [2, 4, 6, 8]

Remove all elements from 'list1' present in 'list2:

[1, 3, 5, 7, 9, 10]

1. Write a Python function to reverse strings in a given list of string values. Original lists:

['Red', 'Green', 'Blue', 'White', 'Black']

Reverse strings of the said given list:

['deR', 'neerG', 'eulB', 'etihW', 'kcalB']

1. Write a Python function to calculate the sum of the numbers in a list between the indices of a specified range.

Original list:

[2, 1, 5, 6, 8, 3, 4, 9, 10, 11, 8, 12]

Range: 8 , 10

Sum of the specified range:

29

1. Write a Python function to count the frequency of consecutive duplicate elements in a given list of numbers.

Original lists:

[1, 2, 2, 2, 4, 4, 4, 5, 5, 5, 5]

Consecutive duplicate elements and their frequency:

([1, 2, 4, 5], [1, 3, 3, 4])

1. Write a Python function to remove duplicate words from a given list of strings.

Original String:

['Python', 'Exercises', 'Practice', 'Solution', 'Exercises']

After removing duplicate words from the said list of strings:

['Python', 'Exercises', 'Practice', 'Solution']

1. Write a Python function to remove empty lists from a given list of lists.

Original list:

[[], [], [], 'Red', 'Green', [1, 2], 'Blue', [], []]

After deleting the empty lists from the said lists of lists

['Red', 'Green', [1, 2], 'Blue']

1. Write a Python function to sum a specific column of a list in a given list of lists.

Original list of lists:

[[1, 2, 3, 2], [4, 5, 6, 2], [7, 8, 9, 5]]

Sum: 1st column of the said list of lists:

12

Sum: 2nd column of the said list of lists:

15

Sum: 4th column of the said list of lists:

9

1. Write a Python function to check if a given element occurs at least n times in a list.

Original list:

[0, 1, 3, 5, 0, 3, 4, 5, 0, 8, 0, 3, 6, 0, 3, 1, 1, 0]

Check if 3 occurs at least 4 times in a list:

True

Check if 0 occurs at least 5 times in a list:

True

Check if 8 occurs at least 3 times in a list:

False

1. Write a Python function to get the index of the first element which is greater than a specified element.

Original list:

[12, 45, 23, 67, 78, 90, 100, 76, 38, 62, 73, 29, 83]

Index of the first element which is greater than 73 in the said list:

4

Index of the first element which is greater than 21 in the said list:

1

Index of the first element which is greater than 80 in the said list:

5

Index of the first element which is greater than 55 in the said list:

3

1. Write a Python function to insert an element in a given list after every nth position.

Original list:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]

Insert a in the said list after 2 nd element:

[1, 2, 'a', 3, 4, 'a', 5, 6, 'a', 7, 8, 'a', 9, 0]

Insert b in the said list after 4 th element:

[1, 2, 3, 4, 'b', 5, 6, 7, 8, 'b', 9, 0]

1. Write a Python function to find common elements in a given list of lists.

Original list:

[[7, 2, 3, 4, 7], [9, 2, 3, 2, 5], [8, 2, 3, 4, 4]]

Common elements of the said list of lists:

[2, 3]

Original list:

[['a', 'b', 'c'], ['b', 'c', 'd'], ['c', 'd', 'e']]

Common elements of the said list of lists:

['c']

1. Write a Python function to convert a given decimal number to binary list.

Original Number: 8

Decimal number ( 8 ) to binary list:

[1, 0, 0, 0]

Original Number: 45

Decimal number ( 45 ) to binary list:

[1, 0, 1, 1, 0, 1]

Original Number: 100

Decimal number ( 100 ) to binary list:

[1, 1, 0, 0, 1, 0, 0]

1. Write a Python function to find the indices of elements of a given list, greater than a specified value. Original list:

[1234, 1522, 1984, 19372, 1000, 2342, 7626]

Indices of elements of the said list, greater than 3000

[3, 6]

Original list:

[1234, 1522, 1984, 19372, 1000, 2342, 7626]

Indices of elements of the said list, greater than 20000

[]

1. Write a Python function to count the number of groups of non-zero numbers separated by zeros of a given list of numbers.

Original list:

[3, 4, 6, 2, 0, 0, 0, 0, 0, 0, 6, 7, 6, 9, 10, 0, 0, 0, 0, 0, 5, 9, 9, 7, 4, 4, 0, 0, 0, 0, 0, 0, 5, 3, 2, 9, 7, 1]

Number of groups of non-zero numbers separated by zeros of the said list: 4

1. Write a Python function to compute the sum of non-zero groups (separated by zeros) of a given list of numbers.

Original list:

[3, 4, 6, 2, 0, 0, 0, 0, 0, 0, 6, 7, 6, 9, 10, 0, 0, 0, 0, 0, 7, 4, 4, 0, 0, 0, 0, 0, 0, 5, 3, 2, 9, 7, 1, 0, 0, 0]

Compute the sum of non-zero groups (separated by zeros) of the said list of numbers: [15, 38, 15, 27]

1. Write a Python function to check if two given lists contain the same elements regardless of order.

Sample Output:

Original list elements:

[1, 2, 4]

[2, 4, 1]

Check two said lists contain the same elements regardless of order!

True

Original list elements:

[1, 2, 3]

[1, 2, 3]

Check two said lists contain the same elements regardless of order!

True

Original list elements:

[1, 2, 3]

[1, 2, 4]

Check two said lists contain the same elements regardless of order!

False

1. Write a function that for a given list of strings, returns a list where each string has all its ”x” removed.

noX([”ax”, ”bb”, ”cx”]) -> [”a”, ”bb”, ”c”]

noX([”xxax”, ”xbxbx”, ”xxcx”]) -> [”a”, ”bb”, ”c”]

noX([”x”]) -> [””]

1. Write a function that counts how many words occur in a list up to and including the first word containing character q or character w.
2. Given an array of ints, return True if 6 appears as either the first or last element in the array. The array will be length 1 or more.  
   first\_last6([1, 2, 6]) → True  
   first\_last6([6, 1, 2, 3]) → True  
   first\_last6([13, 6, 1, 2, 3]) → False
3. Given an array of ints, return True if the array is length 1 or more, and the first element and the last element are equal.

same\_first\_last([1, 2, 3]) → False

same\_first\_last([1, 2, 3, 1]) → True

same\_first\_last([1, 2, 1]) → True

1. Given 2 arrays of ints, a and b, return True if they have the same first element or they have the same last element. Both arrays will be length 1 or more.

common\_end([1, 2, 3], [7, 3]) → True

common\_end([1, 2, 3], [7, 3, 2]) → False

common\_end([1, 2, 3], [1, 3]) → True

1. Given an array of ints length 3, return an array with the elements "rotated left" so {1, 2, 3} yields {2, 3, 1}.

rotate\_left3([1, 2, 3]) → [2, 3, 1]

rotate\_left3([5, 11, 9]) → [11, 9, 5]

rotate\_left3([7, 0, 0]) → [0, 0, 7]

1. Given an array of ints length 3, return a new array with the elements in reverse order, so {1, 2, 3} becomes {3, 2, 1}.

reverse3([1, 2, 3]) → [3, 2, 1]

reverse3([5, 11, 9]) → [9, 11, 5]

reverse3([7, 0, 0]) → [0, 0, 7]

1. Given an array of ints length 3, figure out which is larger, the first or last element in the array, and set all the other elements to be that value. Return the changed array.

max\_end3([1, 2, 3]) → [3, 3, 3]

max\_end3([11, 5, 9]) → [11, 11, 11]

max\_end3([2, 11, 3]) → [3, 3, 3]

1. Given an array of ints, return the sum of the first 2 elements in the array. If the array length is less than 2, just sum up the elements that exist, returning 0 if the array is length 0.

sum2([1, 2, 3]) → 3

sum2([1, 1]) → 2

sum2([1, 1, 1, 1]) → 2

1. Given 2 int arrays, a and b, each length 3, return a new array length 2 containing their middle elements.

middle\_way([1, 2, 3], [4, 5, 6]) → [2, 5]

middle\_way([7, 7, 7], [3, 8, 0]) → [7, 8]

middle\_way([5, 2, 9], [1, 4, 5]) → [2, 4]

1. Given an array of ints, return a new array length 2 containing the first and last elements from the original array. The original array will be length 1 or more.

make\_ends([1, 2, 3]) → [1, 3]

make\_ends([1, 2, 3, 4]) → [1, 4]

make\_ends([7, 4, 6, 2]) → [7, 2]

1. Given an int array length 2, return True if it contains a 2 or a 3.

has23([2, 5]) → True

has23([4, 3]) → True

has23([4, 5]) → False

1. Return the number of even ints in the given array. Note: the % "mod" operator computes the remainder, e.g. 5 % 2 is 1.

count\_evens([2, 1, 2, 3, 4]) → 3

count\_evens([2, 2, 0]) → 3

count\_evens([1, 3, 5]) → 0

1. Given an array length 1 or more of ints, return the difference between the largest and smallest values in the array. Note: the built-in min(v1, v2) and max(v1, v2) functions return the smaller or larger of two values.

big\_diff([10, 3, 5, 6]) → 7

big\_diff([7, 2, 10, 9]) → 8

big\_diff([2, 10, 7, 2]) → 8

1. Return the "centered" average of an array of ints, which we'll say is the mean average of the values, except ignoring the largest and smallest values in the array. If there are multiple copies of the smallest value, ignore just one copy, and likewise for the largest value. Use int division to produce the final average. You may assume that the array is length 3 or more.

centered\_average([1, 2, 3, 4, 100]) → 3

centered\_average([1, 1, 5, 5, 10, 8, 7]) → 5

centered\_average([-10, -4, -2, -4, -2, 0]) → -3

1. Return the sum of the numbers in the array, returning 0 for an empty array. Except the number 13 is very unlucky, so it does not count and numbers that come immediately after a 13 also do not count.

sum13([1, 2, 2, 1]) → 6

sum13([1, 1]) → 2

sum13([1, 2, 2, 1, 13]) → 6

1. Return the sum of the numbers in the array, except ignore sections of numbers starting with a 6 and extending to the next 7 (every 6 will be followed by at least one 7). Return 0 for no numbers.

sum67([1, 2, 2]) → 5

sum67([1, 2, 2, 6, 99, 99, 7]) → 5

sum67([1, 1, 6, 7, 2]) → 4

1. Given an array of ints, return True if the array contains a 2 next to a 2 somewhere.

has22([1, 2, 2]) → True

has22([1, 2, 1, 2]) → False

has22([2, 1, 2]) → False

1. Write a function that computes the sum of all elements in a list up to but not including the first number bigger than 5.

# Dictionaries

1. Write a Python function to concatenate the three dictionaries to create a new one.

Sample Dictionary :

dic1={1:10, 2:20}

dic2={3:30, 4:40}

dic3={5:50,6:60}

Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

1. Write a Python function to check whether a given key already exists in a dictionary.
2. Write a Python function to iterate over dictionaries using for loops.
3. Write a Python function to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x\*x).

Sample Dictionary ( n = 5) :

Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

1. Write a Python function to sum all the values in a dictionary.
2. Write a Python function to multiply all the values in a dictionary.
3. Write a Python function to remove a key from a dictionary.
4. Write a Python function to get the maximum and minimum value in a dictionary.
5. Write a Python function to combine two dictionary adding values for common keys.  
   d1 = {'a': 100, 'b': 200, 'c':300}

d2 = {'a': 300, 'b': 200, 'd':400}

Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})

1. Write a Python function to print all unique values in a dictionary.

Sample Data : [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S005"}, {"V":"S009"},{"VIII":"S007"}]

Expected Output : Unique Values: ('S005', 'S002', 'S007', 'S001', 'S009')

1. Write a Python function to create and display all combinations of letters, selecting each letter from a different key in a dictionary.   
   Sample data : {'1':['a','b'], '2':['c','d']}

Expected Output:

ac

ad

bc  
bd

1. Write a Python function to combine values in a python list of dictionaries.

Sample data: [{'item': 'item1', 'amount': 400}, {'item': 'item2', 'amount': 300}, {'item': 'item1', 'amount': 750}]

Expected Output: Counter({'item1': 1150, 'item2': 300})

1. Write a Python function to remove spaces from dictionary keys.
2. Write a Python function to match key values in two dictionaries.

Sample dictionary: {'key1': 1, 'key2': 3, 'key3': 2}, {'key1': 1, 'key2': 2}

Expected output: key1: 1 is present in both x and y

1. Write a Python function to filter even numbers from a given dictionary values.

Original Dictionary:

{'V': [1, 4, 6, 10], 'VI': [1, 4, 12], 'VII': [1, 3, 8]}

Filter even numbers from said dictionary values:

{'V': [4, 6, 10], 'VI': [4, 12], 'VII': [8]}

Original Dictionary:

{'V': [1, 3, 5], 'VI': [1, 5], 'VII': [2, 7, 9]}

Filter even numbers from said dictionary values:

{'V': [], 'VI': [], 'VII': [2]}

1. Write a Python function to get the total length of all values of a given dictionary with string values.

Original dictionary:

{'#FF0000': 'Red', '#800000': 'Maroon', '#FFFF00': 'Yellow', '#808000': 'Olive'}

Total length of all values of the said dictionary with string values:

20

1. Write a Python function to create a list of all the keys in a flat dictionary.

Sample Output:

Original dictionary elements:

{'Theodore': 19, 'Roxanne': 20, 'Mathew': 21, 'Betty': 20}

Create a list of all the keys of the said flat dictionary:

['Theodore', 'Roxanne', 'Mathew', 'Betty']

1. Write a Python function to create a list of all the values in a flat dictionary.

Sample Output:

Original dictionary elements:

{'Theodore': 19, 'Roxanne': 20, 'Mathew': 21, 'Betty': 20}

Create a flat list of all the values of the said flat dictionary:

[19, 20, 21, 20]

1. Write a Python function to find the key of the maximum value in a dictionary.

Sample Output:

Original dictionary elements:

{'Theodore': 19, 'Roxanne': 22, 'Mathew': 21, 'Betty': 20}

Finds the key of the maximum and minimum value of the said dictionary:

('Roxanne', 'Theodore')

1. Write a Python function to convert a given dictionary to a list of tuples.

Sample Output:

Original Dictionary:

{'Red': 1, 'Green': 3, 'White': 5, 'Black': 2, 'Pink': 4}

Convert the said dictionary to a list of tuples:

[('Red', 1), ('Green', 3), ('White', 5), ('Black', 2), ('Pink', 4)]

1. Write a Python function to combine two lists into a dictionary, where the elements of the first one serve as the keys and the elements of the second one serve as the values. The values of the first list need to be unique and hashable.

Sample Output:

Original lists:

['a', 'b', 'c', 'd', 'e', 'f']

[1, 2, 3, 4, 5]

Combine the values of the said two lists into a dictionary:

{'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}

1. What is printed by the following statements?

thedict = {"aap": 4, "ole": 6, "en":13, "fr": 77}

thedict["rex"] = thedict["aap"] + thedict["fr"]

thedict["cop"] = thedict["ole"] + thedict["en"]

print(thedict["rex"])

for akey in thedict.keys():

print("key", akey, "maps to", thedict[akey])

ahalist = list(thedict.keys())

print(ahalist)

ahalist.sort()

print(ahalist[-2]+ahalist[-1])

1. What is printed by the following statements?

total = 0

thedict = {"ap": 2, "do": 6, "ty":3, "fruit": 17}

for akey in thedict:

if len(akey) > 5:

total = total + thedict[akey]

print(total)

print("rex" in thedict)

print(22 in thedict)

1. What is printed by the following statements?  
   total = 0

thedict = {"apolo": 22, "orlando": 36, "qwerty":23, "notafruit": 17}

for akey in thedict:

if len(akey) > 5:

total = total + thedict[akey]

print(total)

print("charlie" in thedict)

print(23 in thedict)

1. What is printed by the following statements?

total = 0

mydict = {"apple":13, "orange":16, "plum":23, "grapefruit": 17}

for akey in mydict:

if len(akey) > 4:

total = total + mydict[akey]

print(total)

print("orange" in mydict)

print(23 in mydict)

1. What is printed by the following statements?

tvoc = {"qw": 4, "en":13, "fro": 77, "klo": 2}

tvoc ["rex"] = tvoc["klo"] + tvoc["fro"]+ tvoc["en"]

tvoc["str"] = tvoc["fro"] + tvoc["fro"]

print(tvoc["rex"])

for akey in tvoc.keys():

print("key", akey, "maps to", tvoc [akey])

ahalist = list(tvoc .keys())

print(ahalist)

ahalist.sort()

print(ahalist[-2]+ahalist[-1])

1. Write a function that for a given dictionary (whose all values are integers) returns a dictionary in which all values are replaced by their rightmost digit (for example 29 is replaced by 9).
2. Write a function that for a given dictionary (whose all values are integers) returns a dictionary in which all values smaller than 14 are doubled.
3. Write a function that for a given dictionary (whose all values are integers) returns a dictionary in which all values smaller than 9 are doubled.

# Tuples

1. Write a Python function to calculate the product, multiplying all the numbers of a given tuple.

Original Tuple:

(4, 3, 2, 2, -1, 18)

Product - multiplying all the numbers of the said tuple: -864

Original Tuple:

(2, 4, 8, 8, 3, 2, 9)

Product - multiplying all the numbers of the said tuple: 27648

1. Write a Python function to calculate the average value of the numbers in a given tuple of tuples.

Original Tuple:

((10, 10, 10, 12), (30, 45, 56, 45), (81, 80, 39, 32), (1, 2, 3, 4))

Average value of the numbers of the said tuple of tuples:

[30.5, 34.25, 27.0, 23.25]

Original Tuple:

((1, 1, -5), (30, -15, 56), (81, -60, -39), (-10, 2, 3))

Average value of the numbers of the said tuple of tuples:

[25.5, -18.0, 3.75]

1. Write a Python function to convert a tuple of string values to a tuple of integer values.

Original tuple values:

(('333', '33'), ('1416', '55'))

New tuple values:

((333, 33), (1416, 55))

1. Write a Python function to convert a given tuple of positive integers into an integer.

Original tuple:

(1, 2, 3)

Convert the said tuple of positive integers into an integer:

123

Original tuple:

(10, 20, 40, 5, 70)

Convert the said tuple of positive integers into an integer:

102040570

1. Write a Python function to check if a specified element presents in a tuple of tuples.

Original list:

(('Red', 'White', 'Blue'), ('Green', 'Pink', 'Purple'), ('Orange', 'Yellow', 'Lime'))

Check if White present in said tuple of tuples!

True

Check if Olive is present in said tuple of tuples!

False

1. Write a Python function to compute the sum of all the elements of each tuple stored inside a list of tuples.

Original list of tuples:

[(1, 2), (2, 3), (3, 4)]

Sum of all the elements of each tuple stored inside the said list of tuples:

[3, 5, 7]

Original list of tuples:

[(1, 2, 6), (2, 3, -6), (3, 4), (2, 2, 2, 2)]

Sum of all the elements of each tuple stored inside the said list of tuples:

[9, -1, 7, 8]

1. Write a Python function to convert a given list of tuples to a list of lists.

Original list of tuples: [(1, 2), (2, 3), (3, 4)]

Convert the said list of tuples to a list of lists: [[1, 2], [2, 3], [3, 4]]

Original list of tuples: [(1, 2), (2, 3, 5), (3, 4), (2, 3, 4, 2)]

Convert the said list of tuples to a list of lists: [[1, 2], [2, 3, 5], [3, 4], [2, 3, 4, 2]]

# Classes & Objects

1. We can represent a rectangle by knowing three things: the location of its lower left corner, its width, and its height.

* Define a class called Point representing 2D points.
* Create a class definition for a Rectangle class using this idea. To create a Rectangle object at location (1,2) with width 2 and height 3, we would do the following:   
  r = Rectangle(Point(1, 2), 2, 3)
* Add the following accessor methods to the Rectangle class: getWidth, getHeight, setWidth, setHeight, str .
* Write a method in the Rectangle class to test if a Point falls on the diagonal of the rectangle.
* Write a method in the Rectangle class (or a function) to test if two given rectangles have at least one common vertex.
* Write a function (or a method in the Rectangle class) that for two given rectangles returns their intersection (that is it returns a rectangle that is an intersection of the given two rectangles).
* Write a method in the Rectangle class to test if a Point is one of the vertices of the rectangle or its center (the intersection of its diagonals).
* Write a method in the Point class that swaps the coordinates of its lower left corner.
* Write a method in the Rectangle class to test if a Point falls within the rectangle.
* Write a method in the Rectangle class to test if a Point falls on one of the four sides of the rectangle.
* Write a method in the Rectangle class to test if a rectangle is a square.
* Write a method in the Rectangle class to test if a rectangle is intersecting with another rectangle.
* Write a transpose method in the Rectangle class that swaps the width and the height of any rectangle instance.

1. Write a class named Pet, which should have the following data attributes: name, type (example values are ”Dog”, ”Cat”, and ”Lizard”), date of birth, height, weight.

* Create a constructor for this class.
* Create set and get methods for all attributes.
* Create a method that will calculate pet’s age in years for pets older than 2 years, in months for pets younger than 2 years.
* Write a program that creates an object of the class and prompts the user to enter the name, type, height, weight and date of birth of his or her pet. This data should be stored as the object’s attributes.
* Use the object’s methods to retrieve the pet’s name, type, and age and display this data on the screen.

1. Given the following class definition:   
     
     
   write the code for a class named Duck that is a subclass of the Bird class. The Duck class’s constructor should call the Bird class’s constructor, passing ’duck’ as an argument.
2. Given the following class definition:



write the code for a class named Ladybug that is a subclass of the Bug class. The Ladybug class’s constructor should call the Bug class’s constructor, passing ’ladybug’ as an argument.

END OF QUESTIONS