

PYTHON

ASSIGNMENT BOOK



CONSULTADD INC, TRAINING TEAM

@ 2021

TASK THREE

DATA STRUCTURES

1. Create a list of 10 elements of four different data types like int, string, complex and float.
2. Create a list of size 5 and execute the slicing structure
3. Write a program to get the sum and multiply of all the items in a given list.
4. Find the largest and smallest number from a given list.
5. Create a new list which contains the specified numbers after removing the even numbers from a predefined list.
6. Create a list of elements such that it contains the squares of the first and last 5 elements between 1 and 30 (both included).
7. Write a program to replace the last element in a list with another list.



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

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

8. Create a new dictionary by concatenating the following two dictionaries:

Sample input: a={1:10,2:20} b={3:30,4:40}

Expected output: {1:10,2:20,3:30,4:40}

9. Create a dictionary that contain numbers in the form($x:x*x$) where x takes all the values between 1 and n (both 1 and n included).

Sample input: $n=5$

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

10. Write a program which accepts a sequence of comma-separated numbers from console and generates a list and a tuple which contains every number in the form of string.

Sample input: 34,67,55,33,12,98

Expected output: ['34','67','55','33','12','98'] ('34','67','55','33','12','98')



TASK FOUR

TRADITIONAL FUNCTIONS, ANONYMOUS FUNCTIONS & HIGHER ORDER FUNCTIONS

1. Write a program to reverse a string.

Sample input: "1234abcd"

Expected output: "dcba4321"

2. Write a function that accepts a string and prints the number of uppercase letters and lowercase letters.

Sample input: "abcSdefPghijQkl"

Expected Output: No. of Uppercase characters : 3 No. of Lower case Characters : 12

3. Create a function that takes a list and returns a new list with unique elements of the first list.

4. Write a program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.

5. Write a program that accepts a sequence of lines as input and prints the lines after making all characters in the sentence capitalized.

Sample input: Hello world Practice makes man perfect

Expected output: HELLO WORLD PRACTICE MAKES MAN PERFECT

6. Define a function that can receive two integral numbers in string form and compute their sum and print it in the console.

7. Define a function that can accept two strings as input and print the string with the maximum length in the console. If two strings have the same length, then the function should print both the strings line by line.

8. Define a function which can generate and print a tuple where the values are square of numbers between 1 and 20 (both 1 and 20 included).

9. Write a function called showNumbers that takes a parameter called limit. It should print all the numbers between 0 and limit with a label to identify the even and odd numbers.

Sample input: show Numbers(3) (where limit=3)

Expected output:

0 EVEN

1 ODD

2 EVEN

3 ODD



10. Write a program which uses filter() to make a list whose elements are even numbers between 1 and 20 (both included)

11. Write a program which uses map() and filter() to make a list whose elements are squares of even numbers in [1,2,3,4,5,6,7,8,9,10].

Hints: Use filter() to filter even elements of the given list Use map() to generate a list of squares of the numbers in the filtered list. Use lambda() to define anonymous functions.

12. Write a function to compute 5/0 and use try/except to catch the exceptions

13. Flatten the list [1,2,3,4,5,6,7] into 1234567 using reduce().

14. Write a program in Python to find the values which are not divisible by 3 but are a multiple of 7.

Make sure to use only higher order functions.

15. Write a program in Python to multiply the elements of a list by itself using a traditional function and pass the function to map() to complete the operation.

16. What is the output of the following codes:

(i) def foo():

try:

return 1

finally:

return 2

k = foo()

print(k)



(ii) def a():

try:

f(x, 4)

finally:

print('after f')

print('after f?')

a()

Welcome to

INNOVATION WITH PYTHON

Training

Copyright © 2021 Consultadd Inc | All Rights Reserved

