Function Practice Problems

EASY

1. Write a function in Python that will take a string text as input from the user and returns the list of unique characters concatenated with their ASCII value at the front and back side.

Sample Input:

"pythonbook"

Function Calling:

function_name("pythonbook")

Sample Output:

['112p112', '121y121', '116t116', '104h104', '111o111', '110n110', '98b98', '107k107']

2. Write a function in Python that will take a string text as input from the user and returns a dictionary having the unique characters as the keys and the list of their both-way indexes(positive and negative index) as the values.

Sample Input:

"pythonbook"

Function Calling:

function_name("pythonbook")

Sample Output:

{'p': [0, -10], 'y': [1, -9], 't': [2, -8], 'h': [3, -7], 'o': [4, -6, 7, -3, 8, -2], 'n': [5, -5], 'b': [6, -4], 'k': [9, -1]}

MEDIUM

3. Write a function in Python that will take a space separated string text as input from the user and returns a dictionary having the unique words as the keys and their frequency in the given text as the values in a sorted order(ascending) according to the frequencies.

Sample Input:

"go there come and go here and there go care"

Function Calling:

function_name("go there come and go here and there go care")

Sample Output:

{'come': 1, 'here': 1, 'care': 1, 'there': 2, 'and': 2, 'go': 3}

4. Write a function in Python that will take a number string text as input from the user and returns a dictionary having the unique numbers as the keys and the tuple of being the number to be even, odd, prime and perfect as the values.

Sample Input:

"2441396"

Function Calling:

function_name("2441396")

Sample Output:

{2: ('even', 'prime', 'not perfect'), 4: ('even', 'not prime', 'not perfect'), 1: ('odd', 'not prime', 'not perfect'), 3: ('odd', 'prime', 'not perfect'), 9: ('odd', 'not prime', 'not perfect'), 6: ('even', 'not prime', 'perfect')}

5. Write a function in Python that will take two matrices as input from the user in two lists. Then return the summation matrix and print it in the function call. [A matrix can only be added to another matrix if the two matrices have the same dimension]**[Avoid using built-in Functions]**

Sample Input:

matrix_A = [[1,5] , [-4,3]]

matrix_B = [[2,-1] , [4,-1]]

Function Calling:

function_name(matrix_A , matrix_B)

Sample Output:

matrix_sum = [[3,4] , [0,2]]

Explanation:

Inside matrix_A and matrix_B, each list is a row matrix. For example, In matrix_A,
Row 1----->[1,5]

Row 2---->[-4,3]

In matrix_B,Row 1--->[2,-1]

Row 2--->[4,-1]

So in output,matrix_sum = [[1+2 , 5 -1]

[-4+ 4, 4 -1]]

HARD

6. Write a Python program that will take a number N and for the keys(1 to N) take N number of lists-as the values of a dictionary, with N number of elements in each list from the user. Here Key number refers to the row numbers.

Call a **Convert_to _diagonal** function that takes the dictionary of a nondiagonal square matrix. Inside this Function, call another function **convert_to list,** which converts the dictionary into a list of lists and returns the list. Where each list represents a row matrix**.**

After that, convert the square non-diagonal matrix list into a diagonal matrix list. Then print it in the function.**[Avoid using built-in Functions]**

Sample Input:

4

square_matrix_dict = {1 : [1,2,3,4] , 2 : [4,5,6,7] , 3 : [7,8,9,3] , 4:[9,1,2,3] }

Function Calling:

convert_to _diagonal(square_matrix_dict)

Sample Output:

Non Diagonal matrix:

1 2 3 4

4 5 6 7

7 8 9 3

9 1 2 3

Diagonal Matrix :

1 0 0 0

0 5 0 0

0 0 9 0

0 0 0 3

Explanation:

[**Square Matrix**: A square matrix is a matrix with the same number of rows and columns.

And **Diagonal Matrix**: Any given square matrix where all the elements are zero except for the elements that are present diagonally is called a diagonal matrix]

1.Take input from user and create **square_matrix_dict** dictionary and call **convert_to _diagonal(square_matrix_dict)**

2. Convert, square_matrix_dict = {1 : [1,2,3,4] , 2 : [4,5,6,7] , 3 : [7,8,9,3] , 4:[9,1,2,3] }

into a list , nonDiagonal_matrix_list = [[1,2,3,4] , [4,5,6,7] , [7,8,9,3] , [9,1,2,3]] using **convert_to _list** Function

3.Return the list in **convert_to _diagonal** Function and convert the “nonDiagonal_matrix_list” into a “ diagonal_matrix_list”

4. Print both non-diagonal and diagonal matrices like the output.