# ========== Easy ==========

1. Copy the following dictionary in your code and generate the output accordingly.

d= {"First Name":"Sifat", "Last Name":"Tanvir"}

Output:

First Name: Sifat

Last Name: Tanvir

2. Copy the following dictionary in your code and generate the output accordingly using a nested loop.

d= {"A":[10,20,30], "B":(-10,-20,-30), "C":"Junu"}

Output:

10

20

30

-10

-20

-30

J

u

n

u

3. Copy the following dictionary in your code and delete all the even keys ranging from 1-9 (inclusive).

d= {1:"N", 2:"B", 3:"U", 4:"D", 5:"N", 6:"F", 9:"U"}

Output:

{1:"N", 3:"U", 5:"N", 9:"U"}

4. Imagine you have a dictionary like this.

students={"CSE" : ["Alex", "Murphy", "Reynolds"], "BBA": ["Joseph", "Sarah", "Elias"], "MNS":["Ray", "Darby"] }

Now take the name of the department as input from the user and print the corresponding list. If not found in the dictionary, print "Not found".

Sample Input 1:

CSE

Output:

["Alex", "Murphy", "Reynolds"]

Sample Input 2:

LLB

Output:

"Not found"

5. Trace the following code and write the outputs.

```
d1={1:[2,3,4], 2:[4,5,6], 3:[15]}
d2= d1
d3= d2.copy()

for i in d1:
  for j in range(len(d1[i])):
    if(j==0):
      d2[i][j]= d1[i][j]+d3[i][j]
      print(d2[i][j])
    if (j>0):
      d3[i][j]= d1[i][j-1] - d2[i][j]
    else:
      d3[i][j]= d1[i][j] - d2[i][j]
    print(d3[i][j])
```

```
  print(i+d1[i][0]+len(d3))
```

6. Trace the following code and write the outputs.

```
d1={1:[2,3,4], 2:[4,5,6], 3:[15]}
d2= d1
d3= d2.copy()

for i,x in d1.items():
  for j in range(len(x)):
    if (i%3==0):
      d3[i][j]= d1[i][j] - d2[i][j]
    elif (j%2==0):
      d2[i][j]= d1[i][j]+d3[i][j]
    else:
      d3[i][j]= d1[i][j] - d2[i][j]
    print(d3[i][j]+d2[i][j])
  print(i+d1[i][0]+len(d3))
print(d1.keys()== d3.keys())
```

7. Take the following input as a string and convert it into a dictionary using slice/split.

```
d1={1:[2,3,4], 2:[4,5,6], 3:[15]}
```

Output:

```
{1:[2,3,4], 2:[4,5,6], 3:[15]}
```

========== Medium ==========

8. Take 10 inputs from the user and create a dictionary with keys 1 to 5 in such a way that each key will indicate to list of numbers divisible by that key.

Sample Input:

11

2

13

41

25

46

17

28

91

10

Output: { 1 : [11,2,13,41,25,46,17,28,91,10], 2 : [2,46,28,10], 3 : [ ], 4 : [28], 5 : [25, 10] }

9. Merge the following two dictionaries to make the output dictionary:

d1= {"CSE": ["Alex", "Murphy"], "BBA": ["Joseph", "Sarah"] }

d2= {"CSE": ["Murphy", "Reynolds"], "MNS": ["Ray", "Darby"] }

Output:

{"CSE": ["Alex", "Murphy", "Reynolds"], "BBA": ["Joseph", "Sarah"], "MNS": ["Ray", "Darby"] }

10. Imagine you have the following dictionary

d1= { "A": [13, 31, 24], "B": [23, 12, 24], "C":[14, 31] }

You have to create the resultant dictionary by switching the keys and values of the original dictionary so that it looks like this:

Output:

{ 13: ["A"], 31: ["A", "C"], 24: ["A", "B"], 23: ["B"], 12: ["B"], 14: ["C"] }

11. A quote has been given to you. Depending on the results, you have to modify it. Please be aware that your quote is of the string type. To accomplish the goal,

a dictionary must be used. Also keep in mind that the actual input cannot be destroyed. Only the built-in function split() may be used if necessary.

Sample Input 1:

"Good,things,take,time"

Sample Output 1:

{'Good': 'G-o-o-d', 'things': 't-h-i-n-g-s', 'take': 't-a-k-e', 'time': 't-i-m-e'}

Sample Input 2:

“The,right,people,stay”

Sample Output 2:

{'The': 'T-h-e', 'right': 'r-i-g-h-t', 'people': 'p-e-o-p-l-e', 'stay': 's-t-a-y'}

Explanation:

The words in your quote will have hyphens (-) in between each letter. Make sure there are no hyphens (-) at the start or end of the words in your quote.

12. Suppose you are given the following dictionary. Just copy the dictionary directly onto your script.

d1= {"0": (3, 4), "1": (2, 3), "2": (1,2), "3": (1, 0), "5": (1, 6), "6": (1, 5), "7": (2, 1)}

The dictionary is designed in such a way that each key is a digit and indicates a tuple. The first element of the tuple is the frequency of the digit and the second

element is the order/sequence in which the digit (key) will occur.

Write a python program that creates a number using the help of the dictionary so that each key in the dictionary appears in the number according to its order/sequence and frequency.

Output:

37721100065

Explanation:

In the given dictionary d1, the second value of the tuple under key "3" is 0. Therefore, the number 3 will come first. Since, the first value of the same tuple is 1, 3

will occur once in the constructed number. Then, the second value of the tuple under key "7" is 1. Therefore, the number 7 will come next. And since, the first value of the same tuple is 2, 7 will occur twice.

After that the numbers 2, 1, 0, 6 and 5 will occur 1, 2, 3, 1 and 1 times respectively. The process will continue until all the keys have been added to the constructed number.

13. Write a program that will take an input of a simple mathematical expression from the user. It is guaranteed that the user will enter an expression that will have 2 operands,

1 operator (either + or - or *), an equal sign, and a result value, and all the things will be separated by a space(" ") in correct order.

Generate and print a dictionary where the keys- 'Operand 1', 'Operand 2', 'Operator', 'Is Correct?' have their corresponding values.

The value of the key ‘Is Correct?’ will be True if the expression is correct (LHS = RHS), otherwise False. In case of incorrect expressions, add another key named 'Correct Result'

and its value should be the correct result of the given expression.

Sample Input 1:

82 * 92 = 7544

Sample Output 1:

{'Operand 1': 82, 'Operand 2': 92, 'Operator':'*', 'Is Correct?': True}

Sample Input 2:

75 - 23 = 98

Sample Output 2:

{'Operand 1': 75, 'Operand 2': 23, 'Operator':'-', 'Is Correct?': False, 'Correct Result': 52}

14. Write a Python program that takes a string as an input where multiple numbers are

separated by hashes(#). Your first task is to create a list of numbers and print it.

Your second task is to create a dictionary that will have the index of the list as key, and for

the even indices, multiplication from start to that index as value, while for the odd indices,

summation from start to that index as value. Finally, print the dictionary.

==========

Sample Input 1:

1#2#3#4

Sample Output 1:

[1, 2, 3, 4]

{0: 1, 1: 3, 2: 6, 3: 10}

Explanation 1:

key 0 → value 1

key 1 → value 1+2 = 3

key 2 → value 1x2x3 = 6

key 3 → value 1+2+3+4 = 10

Sample Output 2:

[5, 6, 7]

{0: 5, 1: 11, 2: 210}

Explanation 2:

key 0 → value 5

key 1 → value 5+6 = 11

key 2 → value 5x6x7 = 210

15. The coordinates of different persons are given in the following dictionary.

position = {'A': (2, 5), 'B': (1, 7), 'C': (9, 5), 'D': (2, 6)}

You will take 2 inputs for person X's location where the first input specifies person X's current location in the x-axis and the second input specifies person X's current location in the y-axis. Your work is to find the person closest to person X by calculating the distance between person X and every other person in the dictionary.

Hint: The distance between two coordinates is given by the formula (You need to import math):

math.sqrt((x2 - x1)**2 + (y2 - y1)**2)

==========

Sample Input 1:

Enter person X's x coordinate: 3

Enter person X's y coordinate: 6

Sample Output 1:

D

Explanation 1:

X is at (3,6).

The distance between person X and person A: 1.4142

The distance between person X and person B: 2.23606

The distance between person X and person C: 6.08276

The distance between person X and person D: 1.00

The least distance is between person X and D. So, D is selected

==========

Sample Input 2:

Enter person X's x coordinate: 3

Enter person X's y coordinate: 5

Sample Output 2:

A

Explanation 2:

X is at (3,5).

The distance between person X and person A: 1.00

The distance between person X and person B: 2.82842

The distance between person X and person C: 2.44948

The distance between person X and person D: 1.4142

The least distance is between person X and A. So, A is selected