Dictionaries Level - 2

Practice Problem 39

📌 Description:

> - Write a Python program that counts the frequency of each value in a dictionary.
> - The program should create a new dictionary to **map each value** in the original dictionary **to its frequency** (how many times it occurs).
> - If the dictionary is empty, print an empty dictionary.

🔹 Expected Output:

If this is the dictionary:

```
my_dict = {
        "a": 4,
        "b": 4,
        "c": 2,
        "d": 7,
        "e": 4,
        "f": 2,
        "g": 7,
        "h": 7
 }
```

The output should be:

```
freq_dict = {
        4: 3
        2: 2
        7: 3
}
```

Each value in `my_dict` is a **key** in the `freq_dict` and it is mapped to its corresponding frequency as the value.

Practice Problem 40

📌 Description:

> - Write a Python program that creates a dictionary from the values contained in nested lists.
> - Each nested list has this format `[value1, value2]`.
> - `value1` should be the **key** in the dictionary and `value2` should be its corresponding **value**.
> - If there are no nested lists, print an empty dictionary.

🔹 Expected Output:

If this is the list that contains nested lists:

`[["a", 1], ["b", 2], ["c", 3], ["d", 4]]`

The result should be:

`{"a": 1, "b": 2, "c": 3, "d": 4}`

Practice Problem 41

📌 Description:

> - Write a Python program that prints the largest of the values in a dictionary.
> - You may assume that all the values in the dictionary are either lists or tuples.

🔹 Expected Output:

If this is the dictionary:

```
my_dict = {
        "a": [1, 2, 3],
        "b": [4, 0, -4],
        "c": [3, 5, 9],
        "d": [45, 12, 100]
}
```

This should be the output:

`157`

🔸 Hints:

- The `sum()` function returns the sum of the elements of a list or tuple.

Practice Problem 42

📌 Description:

> - Write a Python program that creates and displays a dictionary that maps each **letter** in a string to how many times the character occurs in the string (its frequency).
> - The dictionary should only include the characters in the string.
> - The test should be case-insensitive ("A" should be counted as "a").
> - The keys in the dictionary should be **lowercase** letters.
> - Only include letters in the dictionary.

🔹 Expected Output:

Example 1:

For the string:

`"Hello, World"`

The output should be this dictionary:

`{"h": 1, "e": 1, "l": 3, "o": 2, "w": 1, "r": 1, "d": 1}`

Each letter is mapped to its corresponding frequency.

Example 2:

`"Excellent"`

The output should be this dictionary:

`{"e": 3, "x": 1, "c": 1, "l": 2, "n": 1, "t": 1}`

Practice Problem 43

📌 Description:

> - Write a Python program that sorts (in ascending order) the lists contained as values in a dictionary.
> - The dictionary contains key-value pairs that match strings to lists. You need to sort these lists.
> - The lists have to be **mutated** (changed).

🔹 Expected Output:

If this is the dictionary:

```
my_dict = {
        "a": [4, 3, 2],
        "b": [5, 3, 7],
        "c": [1, 9, 10],
        "d": [3, 4, 1]
}
```

The final output should be:

```
my_dict = {
        "a": [2, 3, 4],
        "b": [3, 5, 7],
        "c": [1, 9, 10],
        "d": [1, 3, 4]
}
```

Notice how all the lists are now sorted in ascending order.

🔸 Hints:

- The `.sort()` method sorts a list (the list is mutated/changed).
- Be careful with using `sorted()` because it only returns a sorted copy of the list.

Practice Problem 44

📌 Description:

> - Write a Python program that takes the content of a dictionary and converts it into a list of lists.
> - Each nested list should contain a key as the first element and its corresponding value as the second element.
> - Print the final list of lists.

🔹 Expected Output:

If this is the original dictionary:

```
product_info = {
        "description": "shoe",
        "price": 4.56,
        "colors": ["green", "blue", "red"],
}
```

The output should be:

'description', 'shoe'], ['price', 4.56], ['colors', ['green', 'blue', 'red']

🔸 Hints:

- The `.items()` dictionary method can be helpful to solve this exercise. It returns a sequence with the keys of the dictionary and their corresponding values.
