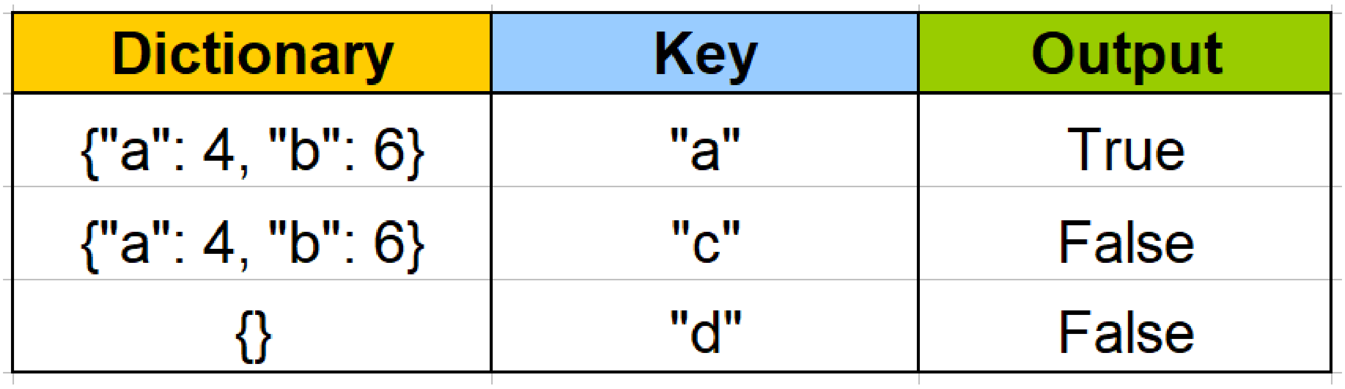
Dictionaries Level - 1

Practice Problem 33

📌 Description:

> - Write a Python program that checks if a key exists in a dictionary or not.
> - If the key exists in the dictionary, print `True`. Else, print `False`.
> - The key should be stored in the variable `key`.

🔹 Expected Output:



Dictionary	Key	Output
{"a": 4, "b": 6}	"a"	True
{"a": 4, "b": 6}	"c"	False
{}	"d"	False

🔸 Hints:

- The `in` operator can be very helpful to solve this challenge.

Practice Problem 34

📌 Description:

> - Write a Python program that adds a new key-value pair to a dictionary only if the key doesn't exist already.
> - If the key-value pair exists in the dictionary, do **not** update the existing value. The dictionary should not be modified in this case.
> - Store the new key in the `new_key` variable and the new value in the `new_value` variable.
> - Print the final value of the dictionary.

🔹 Expected Output:

Example 1: New Pair Added

Initial Dictionary:

`{"January": 45, "February": 56, "March": 67}`

New Key-Value Pair:

`"April": 67`

Output:

`{"January": 45, "February": 56, "March": 67, "April": 67}`

Example 2: No Change

Initial Dictionary:

`{"January": 45, "February": 56, "March": 67}`

New Key-Value Pair:

The key already exists in the dictionary
"January": 67

Output:

{"January": 45, "February": 56, "March": 67}

🔸 Hints:

- You can use the `not in` operator to check if a key is not in a dictionary.

Practice Problem 35

📌 Description:

> - Write a Python program that merges two dictionaries and prints the resulting dictionary.
> - "Merging" the dictionaries involves making a new dictionary with the key-value pairs in both dictionaries.

🔹 Expected Output:

From these two dictionaries:

```
my_dict1 = {"a": 1, "b": 2, "c": 3}
my_dict2 = {"c": 4, "d": 6, "e": 8}
```

The output should be:

`final_dict = {'a': 1, 'b': 2, 'c': 4, 'd': 6, 'e': 8}`

🔸 Hints:

- Notice that the key-value pairs that share the same key on both dictionaries are **not** repeated. They are updated with the value of the second dictionary.
- There is a Python operator that you can use to merge two or more dictionaries.

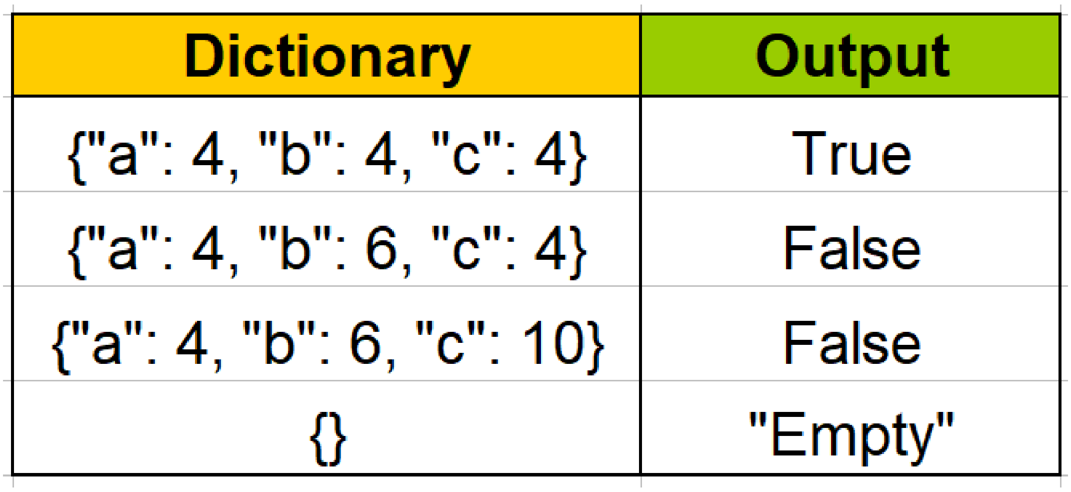
Practice Problem 36

📌 Description:

> - Write a Python program that checks if **all values** in a dictionary are equal.
> - If they are, print `True`. Else, print `False`.

> - If the dictionary is empty, print `"Empty"`.

🔹 **Expected Output:**



Dictionary	Output
{"a": 4, "b": 4, "c": 4}	True
{"a": 4, "b": 6, "c": 4}	False
{"a": 4, "b": 6, "c": 10}	False
{}	"Empty"

🔸 **Hints:**

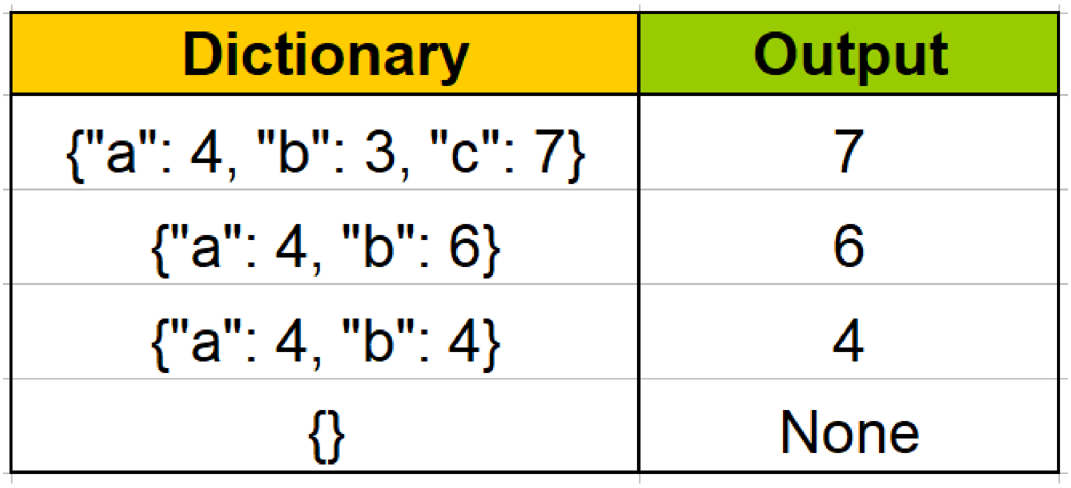
- Loops will be very helpful to solve this challenge. You could also make a set from the dictionary to keep only one copy per value.
- The `.values()` method returns the values in a dictionary.

Practice Problem 37

📌 **Description:**

> - Write a Python program that prints the **largest value** in a dictionary.
> - If the dictionary is empty, print `None`.
> - You may assume that the values are numeric.

🔹 **Expected Output:**



Dictionary	Output
{"a": 4, "b": 3, "c": 7}	7
{"a": 4, "b": 6}	6
{"a": 4, "b": 4}	4
{}	None

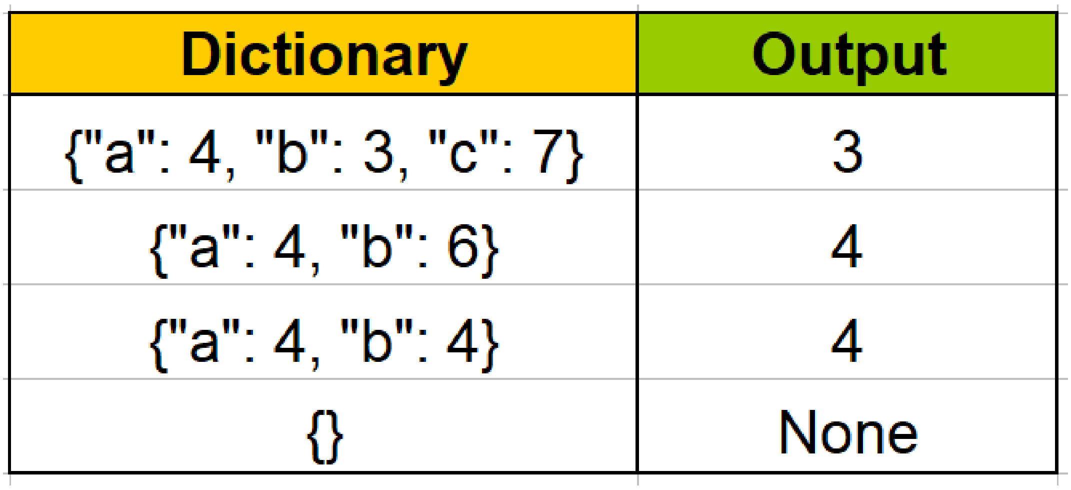
You may print "None" as a string or as the special value None.

Practice Problem 38

📌 Description:

> - Write a Python program that prints the **smallest** value in a dictionary.
> - If the dictionary is empty, print `None`.
> - You may assume that the values are numeric.

🔹 Expected Output:



Dictionary	Output
{"a": 4, "b": 3, "c": 7}	3
{"a": 4, "b": 6}	4
{"a": 4, "b": 4}	4
{}	None

You may print "None" as a string or as the special value None.
