1. **Create a Map**:

Write a Dart program that creates an empty map and then adds at least three key-value pairs to it. Print out the map to verify that it contains the expected data.

2. **Access Map Values**:

Create a map that represents a person's information, such as name, age, and city. Write Dart code to access and print each of these values from the map.

3. **Update Map Values**:

Take the map from the previous task and update the person's age and city. Print the updated map to confirm the changes.

4. **Check for Key Existence**:

Write Dart code that checks if a specific key exists in a map. Create a map, and then check if a given key is present in that map. Print "Key exists" or "Key does not exist" accordingly.

5. **Iterate Over Map**:

Create a map that represents a dictionary of words and their meanings. Write Dart code to iterate over the map and print each word along with its meaning.

6. **Remove Key-Value Pairs**:

Create a map representing a shopping list with items as keys and quantities as values. Write code to remove an item from the list by its key. Print the updated list after removal.

7. **Map** **Length**:

Create a map with a variable number of key-value pairs. Write Dart code to find and print the number of key-value pairs in the map.

8. **Convert** **to** **List**:

Take a map with string keys and integer values. Write Dart code to convert the map into a list of key-value pairs, where each pair is represented as a list or tuple. Print the resulting list.

9. **Merge Maps**:

Create two maps, and then write Dart code to merge them into a single map. If there are overlapping keys, decide how to handle them (e.g., choose one value or add them together). Print the merged map.

10. **Sorting Maps**:

Create a map with string keys and integer values. Write Dart code to sort the map by keys in alphabetical order and then by values in ascending order. Print the sorted map.