Problem 1: Basic Dictionary Creation

Description: Create a dictionary from a list of keys and a list of values. Each key should correspond to a value in the new dictionary.

Input:

python
Copy code
```
keys = ["name", "age", "city"]
values = ["Alice", 30, "New York"]
```

Output:

python
Copy code
```
{"name": "Alice", "age": 30, "city": "New York"}
```

Problem 2: Access Dictionary Values

Description: Given a dictionary and a key, print the value associated with the key.

Input:

python
Copy code
```
data = {"name": "Bob", "age": 25}
key = "name"
```

Output:

python
Copy code
```
"Bob"
```

Problem 3: Update Dictionary Value

Description: Update a value in a dictionary based on a given key.

Input:

python
Copy code
```
data = {"name": "Alice", "age": 30}
key = "age"
new_value = 31
```

Output:

python
Copy code
```
{"name": "Alice", "age": 31}
```

Problem 4: Check Key Existence

Description: Check if a specified key exists in a dictionary. Return `True` or `False`.

Input:

python
Copy code
```
data = {"name": "Alice", "age": 30}
key = "city"
```

Output:

python
Copy code
```
False
```

Problem 5: Remove a Key from Dictionary

Description: Given a dictionary and a key, remove the key-value pair from the dictionary.

Input:

python
Copy code

```
data = {"name": "Alice", "age": 30, "city": "New York"}
key = "age"
```

Output:

python
Copy code
```
{"name": "Alice", "city": "New York"}
```

Problem 6: Count Occurrences in List using Dictionary

Description: Given a list of integers, create a dictionary where keys are the unique integers, and values are the count of their occurrences.

Input:

python
Copy code
```
nums = [1, 2, 2, 3, 1, 1, 4]
```

Output:

python
Copy code
```
{1: 3, 2: 2, 3: 1, 4: 1}
```

Problem 7: Sum of Dictionary Values

Description: Calculate the sum of all values in a dictionary.

Input:

python
Copy code
```
data = {"a": 100, "b": 200, "c": 300}
```

Output:

python
Copy code
```
600
```

Problem 8: Merge Two Dictionaries

Description: Given two dictionaries, merge them into one.

Input:

python
Copy code
```
dict1 = {"a": 1, "b": 2}
dict2 = {"b": 3, "c": 4}
```

Output:

python
Copy code
```
{"a": 1, "b": 3, "c": 4}
```

Problem 9: Nested Dictionary Lookup

Description: Given a nested dictionary, access a specific nested value.

Input:

python
Copy code
```
data = {"info": {"name": "Alice", "age": 30}}
key1 = "info"
key2 = "age"
```

Output:

python
Copy code
```
30
```

Problem 10: Create Dictionary from Two Lists

Description: Create a dictionary where each key-value pair corresponds to the respective elements of two given lists.

Input:

python
Copy code
```
keys = ["a", "b", "c"]
values = [1, 2, 3]
```

Output:

python
Copy code
```
{"a": 1, "b": 2, "c": 3}
```

Problem 11: Reverse Dictionary

Description: Given a dictionary, create a new dictionary by reversing the keys and values.

Input:

python
Copy code
```
data = {"a": 1, "b": 2, "c": 3}
```

Output:

python
Copy code
```
{1: "a", 2: "b", 3: "c"}
```

Problem 12: Filter Dictionary by Value

Description: Given a dictionary and a threshold, create a new dictionary containing only the items with values greater than the threshold.

Input:

python
Copy code
```
data = {"apple": 5, "banana": 3, "cherry": 8}
threshold = 4
```

Output:

python
Copy code
```
{"apple": 5, "cherry": 8}
```

Problem 13: Frequency Count of Characters

Description: Count the frequency of each character in a given string and store it in a dictionary.

Input:

python
Copy code
```
text = "hello"
```

Output:

python
Copy code
```
{"h": 1, "e": 1, "l": 2, "o": 1}
```

Problem 14: Find Maximum Value in Dictionary

Description: Find the key with the maximum value in a dictionary.

Input:

python
Copy code
```
data = {"apple": 5, "banana": 3, "cherry": 8}
```

Output:

python
Copy code
```
"cherry"
```

Problem 15: Dictionary Comprehension for Squares

Description: Use dictionary comprehension to create a dictionary where keys are numbers from 1 to `n`, and values are their squares.

Input:

python
Copy code
```
n = 5
```

Output:

python
Copy code
```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

Problem 16: Update Nested Dictionary Value

Description: Given a nested dictionary and keys, update the nested dictionary with a new value.

Input:

python
Copy code
```
data = {"info": {"name": "Alice", "age": 30}}
key1 = "info"
```

```
key2 = "age"
new_value = 31
```

Output:

python
Copy code
```
{"info": {"name": "Alice", "age": 31}}
```

Problem 17: Dictionary from List of Tuples

Description: Given a list of tuples, convert it into a dictionary.

Input:

python
Copy code
```
pairs = [("name", "Alice"), ("age", 25)]
```

Output:

python
Copy code
```
{"name": "Alice", "age": 25}
```

Problem 18: Sum of Product of Corresponding Values

Description: Given two dictionaries with the same keys, calculate the sum of the product of values with the same keys.

Input:

python
Copy code
```
dict1 = {"a": 2, "b": 3, "c": 4}
dict2 = {"a": 5, "b": 6, "c": 7}
```

Output:

python
Copy code
```
56  # Explanation: (2*5) + (3*6) + (4*7)
```

Problem 19: Count Words in a Sentence

Description: Count the frequency of each word in a given sentence and store it in a dictionary.

Input:

python
Copy code
```
sentence = "hello world hello"
```

Output:

python
Copy code
```
{"hello": 2, "world": 1}
```

Problem 20: Identify Common Keys

Description: Given two dictionaries, return a list of keys common to both.

Input:

python
Copy code
```
dict1 = {"a": 1, "b": 2, "c": 3}
dict2 = {"b": 4, "c": 5, "d": 6}
```

Output:

python
Copy code
```
["b", "c"]
```

Problem 21: Unique Values in a Dictionary

Description: Given a dictionary, return a list of unique values.

Input:

python
Copy code
```
data = {"a": 1, "b": 2, "c": 1, "d": 3}
```

Output:

python
Copy code
```
[1, 2, 3]
```

Problem 22: Find Keys with Minimum Value

Description: Find all keys in a dictionary with the minimum value.

Input:

python
Copy code
```
data = {"a": 2, "b": 1, "c": 3, "d": 1}
```

Output:

python
Copy code
```
["b", "d"]
```

Problem 23: Group Elements by Frequency

Description: Given a list, create a dictionary that groups elements by their frequency.

Input:

python

Copy code
```
elements = [1, 2, 2, 3, 1, 1, 4]
```

Output:

python
Copy code
```
{3: [1], 2: [2], 1: [3, 4]}
```

Problem 24: Flatten a Nested Dictionary

Description: Flatten a nested dictionary. Assume only one level of nesting.

Input:

python
Copy code
```
data = {"a": 1, "b": {"c": 2, "d": 3}}
```

Output:

python
Copy code
```
{"a": 1, "c": 2, "d": 3}
```

Problem 25: Create Dictionary from Two Lists with Different Lengths

Description: Given two lists of keys and values, create a dictionary. If the lengths differ, fill the remaining keys with `None` or ignore extra values.

Input:

python
Copy code
```
keys = ["a", "b", "c"]
values = [1, 2]
```

Output:

python
Copy code
```
{"a": 1, "b": 2, "c": None}
```

Problem 26: Remove Items with Specific Value

Description: Remove all items from a dictionary that have a specific value.

Input:

python
Copy code
```
data = {"a": 1, "b": 2, "c": 1, "d": 3}
value_to_remove = 1
```

Output:

python
Copy code
```
{"b": 2, "d": 3}
```

Problem 27: Dictionary Intersection

Description: Given two dictionaries, create a new dictionary containing only the items with matching keys and values in both dictionaries.

Input:

python
Copy code
```
dict1 = {"a": 1, "b": 2, "c": 3}
dict2 = {"a": 1, "b": 4, "c": 3}
```

Output:

python
Copy code
```
{"a": 1, "c": 3}
```

Problem 28: Sort Dictionary by Value

Description: Sort a dictionary by its values in ascending order and return the sorted dictionary.

Input:

python
Copy code
```
data = {"apple": 5, "banana": 3, "cherry": 8}
```

Output:

python
Copy code
```
{"banana": 3, "apple": 5, "cherry": 8}
```

Problem 29: Count Letters in a List of Words

Description: Given a list of words, create a dictionary where each letter is a key, and the value is the total count of that letter in all words.

Input:

python
Copy code
```
words = ["apple", "banana"]
```

Output:

python
Copy code
```
{"a": 5, "p": 2, "l": 1, "e": 1, "b": 1, "n": 2}
```

Problem 30: Map List Items to Positions

Description: Given a list, create a dictionary where keys are the unique elements of the list, and values are lists of positions (indices) where each element occurs.

Input:

python
Copy code
```
elements = [1, 2, 1, 3, 2, 1]
```

Output:

python
Copy code
```
{1: [0, 2, 5], 2: [1, 4], 3: [3]}
```

Problem 31: Calculate Average of Dictionary Values

Description: Given a dictionary where keys are names and values are scores, calculate the average score.

Input:

python
Copy code
```
scores = {"Alice": 90, "Bob": 80, "Charlie": 70}
```

Output:

python
Copy code
```
80.0
```

Problem 32: Group Dictionary Keys by Value

Description: Given a dictionary, create a new dictionary grouping keys by their values.

Input:

python
Copy code
```
data = {"a": 1, "b": 2, "c": 1, "d": 3}
```

Output:

python
Copy code
```
{1: ["a", "c"], 2: ["b"], 3: ["d"]}
```

Problem 33: Dictionary Difference

Description: Given two dictionaries, find keys that are only in the first dictionary but not in the second.

Input:

python
Copy code
```
dict1 = {"a": 1, "b": 2, "c": 3}
dict2 = {"b": 2, "d": 4}
```

Output:

python
Copy code
```
["a", "c"]
```

Problem 34: Count Vowels in a String

Description: Count the frequency of each vowel in a string and store it in a dictionary.

Input:

python
Copy code
```
text = "hello world"
```

Output:

python
Copy code
```
{"e": 1, "o": 2}
```

Problem 35: Convert Two Lists into a Dictionary with Sum Values

Description: Given two lists, create a dictionary where each unique element in both lists is a key, and the value is the sum of occurrences across both lists.

Input:

python
Copy code
```
list1 = ["a", "b", "a", "c"]
list2 = ["b", "a", "b", "d"]
```

Output:

python
Copy code
```
{"a": 3, "b": 3, "c": 1, "d": 1}
```

Problem 36: Calculate Grades Based on Score Ranges

Description: Given a dictionary of student scores, create a new dictionary categorizing students as "A", "B", "C", or "F" based on score ranges.

- "A": 90-100
- "B": 80-89
- "C": 70-79
- "F": 0-69

Input:

python
Copy code
```
scores = {"Alice": 85, "Bob": 92, "Charlie": 67, "David": 73}
```

Output:

python
Copy code

```
{"Alice": "B", "Bob": "A", "Charlie": "F", "David": "C"}
```

Problem 37: Invert a Dictionary with Lists

Description: Given a dictionary where values are lists, invert it so that each item in the list becomes a key pointing to the original key.

Input:

python
Copy code
```
data = {"fruits": ["apple", "banana"], "vegetables": ["carrot", "beet"]}
```

Output:

python
Copy code
```
{"apple": "fruits", "banana": "fruits", "carrot": "vegetables", "beet": "vegetables"}
```

Problem 38: Filter Dictionary by Key Length

Description: Filter a dictionary to only include keys of a specified length.

Input:

python
Copy code
```
data = {"apple": 5, "banana": 3, "pear": 2}
key_length = 5
```

Output:

python
Copy code
```
{"apple": 5, "pear": 2}
```

Problem 39: Dictionary with Count of Elements Less than K

Description: Given a dictionary with numeric values, create a new dictionary with the count of values less than `k`.

Input:

python
Copy code
```
data = {"a": 5, "b": 2, "c": 7, "d": 3}
k = 4
```

Output:

python
Copy code
```
{"count": 2}
```

Problem 40: Create Dictionary of Squares for Odd Numbers

Description: Given a list of numbers, create a dictionary where each odd number is a key and its value is the square of the number.

Input:

python
Copy code
```
numbers = [1, 2, 3, 4, 5]
```

Output:

python
Copy code
```
{1: 1, 3: 9, 5: 25}
```

Problem 41: Create Frequency Dictionary of Word Lengths

Description: Given a list of words, create a dictionary where each key is the length of a word, and the value is the number of words with that length.

Input:

python
Copy code
```
words = ["apple", "banana", "pear", "kiwi", "orange"]
```

Output:

python
Copy code
```
{5: 2, 6: 1, 4: 1, 7: 1}
```

Problem 42: Create a Dictionary of Divisors

Description: Given a list of numbers, create a dictionary where each number is a key, and the value is a list of all divisors of that number.

Input:

python
Copy code
```
numbers = [10, 15, 20]
```

Output:

python
Copy code
```
{10: [1, 2, 5, 10], 15: [1, 3, 5, 15], 20: [1, 2, 4, 5, 10, 20]}
```

Problem 43: Character Position Mapping

Description: Given a string, create a dictionary where each character is a key, and the value is a list of all positions (indices) where that character appears in the string.

Input:

python
Copy code
```
text = "hello world"
```

Output:

python
Copy code
```
{"h": [0], "e": [1], "l": [2, 3, 9], "o": [4, 7], "w": [6], "r": [8], "d": [10]}
```

Problem 44: Transform Values Based on Function

Description: Given a dictionary of numbers, apply a function (e.g., square root) to each value and create a new dictionary with transformed values.

Input:

python
Copy code
```
data = {"a": 4, "b": 9, "c": 16}
```

Output:

python
Copy code
```
{"a": 2.0, "b": 3.0, "c": 4.0}
```

Problem 45: Count Digits and Letters in String

Description: Given a string, create a dictionary where keys are "letters" and "digits," and values are the count of alphabetic and numeric characters in the string, respectively.

Input:

python
Copy code
```
text = "hello123"
```

Output:

python
Copy code
```
{"letters": 5, "digits": 3}
```

Problem 46: Map List of Numbers to Whether Prime

Description: Given a list of numbers, create a dictionary where each number is a key, and the value is a boolean indicating if the number is prime.

Input:

python
Copy code
```
numbers = [2, 3, 4, 5, 6]
```

Output:

python
Copy code
```
{2: True, 3: True, 4: False, 5: True, 6: False}
```

Problem 47: Convert List of Strings to Nested Dictionary

Description: Given a list of strings, create a nested dictionary structure where each level of nesting corresponds to each character in the string.

Input:

python
Copy code
```
words = ["cat", "car", "dog"]
```

Output:

python
Copy code
```
{'c': {'a': {'t': None, 'r': None}}, 'd': {'o': {'g': None}}}
```

Problem 48: Common Values Across Multiple Dictionaries

Description: Given a list of dictionaries, return a dictionary containing only the key-value pairs that are common to all dictionaries.

Input:

python
Copy code
```
dicts = [{"a": 1, "b": 2}, {"a": 1, "c": 3}, {"a": 1, "b": 2, "c": 3}]
```

Output:

python
Copy code
```
{"a": 1}
```

Problem 49: Sum of Values for Common Keys in List of Dictionaries

Description: Given a list of dictionaries, return a dictionary where each key is a common key, and the value is the sum of values across all dictionaries.

Input:

python
Copy code
```
dicts = [{"a": 1, "b": 2}, {"a": 2, "b": 3, "c": 4}, {"a": 3, "b": 4}]
```

Output:

python
Copy code
```
{"a": 6, "b": 9}
```

Problem 50: Frequency of Sum of Key-Value Pairs in List of Dictionaries

Description: Given a list of dictionaries with numeric key-value pairs, return a dictionary where keys are the sums of values in each dictionary and values are the frequency of these sums.

Input:

python
Copy code
```
dicts = [{"a": 1, "b": 2}, {"a": 2, "b": 2}, {"a": 3, "b": 1}]
```

Output:

python
Copy code
```
{3: 1, 4: 2}
```