**DAY 8 PRACTICE PROGRAMS**

1. **Write a Python script to sort (ascending and descending) a dictionary by value.**

My\_dict = {

'apple': 10,

'banana': 5,

'cherry': 20,

'date': 15

}

sorted\_asc = dict(sorted(my\_dict.items(), key=lambda item: item[1]))

print("Ascending order:", sorted\_asc)

sorted\_desc = dict(sorted(my\_dict.items(), key=lambda item: item[1], reverse=True))

print("Descending order:", sorted\_desc)

**OUTPUT:** Ascending order: {'banana': 5, 'apple': 10, 'date': 15, 'cherry': 20}

Descending order: {'cherry': 20, 'date': 15, 'apple': 10, 'banana': 5}

**2.** **Write a Python program to map two lists into a dictionary.** keys = ['name', 'age', 'city']

values = ['Alice', 30, 'New York']

mapped\_dict = dict(zip(keys, values))

print("Mapped Dictionary:", mapped\_dict)

**OUTPUT:** Mapped Dictionary: {'name': 'Alice', 'age': 30, 'city': 'New York'}

**3.Write a Python program to combine two dictionary adding values for common keys.  
d1 = {'a': 100, 'b': 200, 'c':300}  
d2 = {'a': 300, 'b': 200, 'd':400}  
Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})**

from collections import Counter

d1 = {'a': 100, 'b': 200, 'c': 300}

d2 = {'a': 300, 'b': 200, 'd': 400}

result = Counter(d1) + Counter(d2)

print(result)

**output:** Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})

**4. Using a list comprehension, create a new list that contains only the even numbers from an existing list of integers.**

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

even\_numbers = [num for num in numbers if num % 2 == 0]

print(even\_numbers)

**output:** [2, 4, 6, 8, 10]

**5. Using a list comprehension, create a new list that contains the squares of all the numbers in an existing list.**

numbers = [1, 2, 3, 4, 5]

squares = [num \*\* 2 for num in numbers]

print(squares)

**output:** [1, 4, 9, 16, 25]

**6. Using a list comprehension, create a new set that contains the unique words from an existing list of strings.**

words = ["apple", "banana", "apple", "orange", "banana", "grape"]

unique\_words = {word for word in words}

print(unique\_words)

**output**: {'apple', 'banana', 'orange', 'grape'}

**7. Using a list comprehension, create a new dictionary that contains the frequency of each word in an existing list of strings.**

words = ["apple", "banana", "apple", "orange", "banana", "grape", "apple"]

word\_freq = {word: words.count(word) for word in set(words)}

print(word\_freq)

**output**: {'grape': 1, 'banana': 2, 'orange': 1, 'apple': 3}

**8. Using a list comprehension, create a new list that contains only the elements that are common to two existing lists.**

list1 = [1, 2, 3, 4, 5]

list2 = [4, 5, 6, 7, 8]

common\_elements = [item for item in list1 if item in list2]

print(common\_elements)

**output**: [4, 5]

**9. Using a list comprehension, create a new list that contains the uppercase versions of all the elements in an existing list of strings.**

words = ["apple", "banana", "grape", "orange"]

uppercase\_words = [word.upper() for word in words]

print(uppercase\_words)

**output**: ['APPLE', 'BANANA', 'GRAPE', 'ORANGE']

**10. Using a list comprehension, create a new tuple that contains the elements of an existing list in reverse order.**

numbers = [1, 2, 3, 4, 5]

reversed\_tuple = tuple([numbers[i] for i in range(len(numbers) - 1, -1, -1)])

print(reversed\_tuple)

**output**: (5, 4, 3, 2, 1)

**11. Using a list comprehension, create a new list that contains the elements of an existing list, but with duplicates removed.**

items = [1, 2, 2, 3, 4, 4, 5, 1]

seen = set()

unique\_items = [x for x in items if not (x in seen or seen.add(x))]

print(unique\_items)

**output:** [1, 2, 3, 4, 5]

**12. Using a list comprehension, create a new dictionary that maps the elements of an existing list to their corresponding indices.**

items = ['a', 'b', 'c', 'd']

index\_map = {item: index for index, item in enumerate(items)}

print(index\_map)

**output**: {'a': 0, 'b': 1, 'c': 2, 'd': 3}

**13. Using a list comprehension, create a new list that contains the Cartesian product of two existing lists.**

list1 = [1, 2]

list2 = ['a', 'b']

cartesian\_product = [(x, y) for x in list1 for y in list2]

print(cartesian\_product)

**output:** [(1, 'a'), (1, 'b'), (2, 'a'), (2, 'b')]

**14. How can you use list comprehension to add two matrices:**

matrix1 = [

[1, 2, 3],

[4, 5, 6]

]

matrix2 = [

[7, 8, 9],

[10, 11, 12]

]

result = [[matrix1[i][j] + matrix2[i][j] for j in range(len(matrix1[0]))] for i in range(len(matrix1))]

print(result)

**output**: [[8, 10, 12], [14, 16, 18]]