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

ANS:

def sort\_dictionary\_by\_value(dictionary, ascending=True):

sorted\_dict = dict(sorted(dictionary.items(), key=lambda item: item[1], reverse=not ascending)) return sorted\_dict

input\_dict = {}

num\_items = int(input("Enter the number of items in the dictionary: ")) for i in range(num\_items):

key = input("Enter key: ") value = input("Enter value: ") input\_dict[key] = value

ascending\_sorted\_dict = sort\_dictionary\_by\_value(input\_dict) print("Ascending order:", ascending\_sorted\_dict)

descending\_sorted\_dict = sort\_dictionary\_by\_value(input\_dict, ascending=False) print("Descending order:", descending\_sorted\_dict)

# Write a Python script to add a key to a dictionary. Sample Dictionary : {0: 10, 1: 20}

Expected Result : {0: 10, 1: 20, 2: 30}

# ANS

sample\_dict = {0: 10, 1: 20}

# new\_key = 2

new\_value = 30 sample\_dict[new\_key] = new\_value print(sample\_dict)

# Write a Python script to concatenate following dictionaries to create a new one. Sample Dictionary :

dic1={1:10, 2:20}

# dic2={3:30, 4:40}

dic3={5:50,6:60}

# Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

ANS:

dic1 = {1: 10, 2: 20}

dic2 = {3: 30, 4: 40}

dic3 = {5: 50, 6: 60}

result = {} result.update(dic1) result.update(dic2) result.update(dic3) print(result)

# Write a Python script to check if a given key already exists in a dictionary. ANS:

my\_dict = {'a': 1, 'b': 2, 'c': 3}

def check\_key\_existence(key, dictionary): if key in dictionary:

return True else:

return False

key\_to\_check = input("Enter the key to check: ") if check\_key\_existence(key\_to\_check, my\_dict):

print("The key '{}' exists in the dictionary.".format(key\_to\_check)) else:

print("The key '{}' does not exist in the dictionary.".format(key\_to\_check))

# Write a Python program to iterate over dictionaries using for loops.

ANS:

my\_dict = {'a': 1, 'b': 2, 'c': 3}

print("Keys:")

for key in my\_dict: print(key)

print("\nValues:")

for value in my\_dict.values(): print(value)

print("\nKey-Value Pairs:")

for key, value in my\_dict.items():

print("Key: {}, Value: {}".format(key, value))

# Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x\*x).

Sample Dictionary ( n = 5) :

# Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25} ANS:

def squared\_dict(n): squared\_dict = {}

for x in range(1, n+1): squared\_dict[x] = x \* x

return squared\_dict n = 5

result\_dict = squared\_dict(n) print(result\_dict)

# Write a Python script to merge two Python dictionaries. ANS:

def merge\_dicts(dict1, dict2): result\_dict = dict1.copy() result\_dict.update(dict2) return result\_dict

dict1\_input = input("Enter the first dictionary in the format {key1: value1, key2: value2, ...}: ") dict1 = eval(dict1\_input)

dict2\_input = input("Enter the second dictionary in the format {key1: value1, key2: value2, ...}: ") dict2 = eval(dict2\_input)

merged\_dict = merge\_dicts(dict1, dict2)

print("Merged Dictionary:") print(merged\_dict)

# Write a Python program to sum all the items in a dictionary.

ANS:

my\_dict = {'a': 10, 'b': 20, 'c': 30}

def sum\_items(dictionary): total = 0

for value in dictionary.values(): total += value

return total

result = sum\_items(my\_dict)

print("Sum of all items in the dictionary:", result)

# Write a Python program to multiply all the items in a dictionary. ANS:

my\_dict = {'a': 2, 'b': 3, 'c': 4}

def multiply\_items(dictionary): result = 1

for value in dictionary.values(): result \*= value

return result

result = multiply\_items(my\_dict)

print("Multiplication of all items in the dictionary:", result)

# Write a Python program to remove a key from a dictionary. ANS:

dict1\_input = input("Enter the dictionary in the format {key1: value1, key2: value2, ...}: ") dict1 = eval(dict1\_input)

key\_to\_remove = input("Enter the key to remove from the dictionary: ") if key\_to\_remove in dict1:

del dict1[key\_to\_remove]

print("Key removed successfully.") print("Updated dictionary:", dict1)

else:

print("The key '{}' does not exist in the dictionary.".format(key\_to\_remove))

# Write a Python program to sort a dictionary by key. ANS:

def sort\_dict\_by\_key(dictionary):

sorted\_dict = dict(sorted(dictionary.items(), key=lambda item: item[0])) return sorted\_dict

dict\_input = input("Enter the dictionary in the format {key1: value1, key2: value2, ...}: ") my\_dict = eval(dict\_input)

sorted\_dict = sort\_dict\_by\_key(my\_dict)

print("Sorted dictionary by key:") print(sorted\_dict)

# Write a Python program to get the maximum and minimum value in a dictionary. ANS:

def get\_max\_min\_values(dictionary):

# max\_value = max(dictionary.values(), default=None) min\_value = min(dictionary.values(), default=None) return max\_value, min\_value

dict\_input = input("Enter the dictionary in the format {key1: value1, key2: value2, ...}:") my\_dict = eval(dict\_input)

# max\_value, min\_value = get\_max\_min\_values(my\_dict)

print("Maximum value in the dictionary:", max\_value) print("Minimum value in the dictionary:", min\_value

# Write a Python program to remove duplicates from Dictionary. ANS:

def remove\_duplicates(dictionary): unique\_dict = {}

for key, value in dictionary.items():

if value not in unique\_dict.values(): unique\_dict[key] = value

return unique\_dict

dict\_input = input("Enter the dictionary in the format {key1: value1, key2: value2, ...}: ") my\_dict = eval(dict\_input)

unique\_dict = remove\_duplicates(my\_dict)

print("Dictionary without duplicates:") print(unique\_dict)

# Write a Python program to check a dictionary is empty or not. ANS:

def is\_dict\_empty(dictionary): return not bool(dictionary)

dict\_input = input("Enter the dictionary in the format {key1: value1, key2: value2, ...}: ") my\_dict = eval(dict\_input)

if is\_dict\_empty(my\_dict): print("The dictionary is empty.")

else:

print("The dictionary is not empty.")

# 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

# def combine\_dicts(dict1, dict2):

# """

# Combines two dictionaries adding values for common keys.

# Args:

# dict1 (dict): The first dictionary.

# dict2 (dict): The second dictionary.

# Returns:

# A new dictionary with the combined values.

# """

# # Create a new dictionary with the combined values.

# new\_dict = {}

# # Iterate over the keys in the first dictionary.

# for key in dict1:

# # If the key is in the second dictionary, add the values together.

# if key in dict2:

# new\_dict[key] = dict1[key] + dict2[key]

# # Otherwise, just add the value from the first dictionary.

# else:

# new\_dict[key] = dict1[key]

# # Iterate over the keys in the second dictionary.

# for key in dict2:

# # If the key is not in the first dictionary, add it to the new dictionary.

# if key not in dict1:

# new\_dict[key] = dict2[key]

# # Return the new dictionary.

# return new\_dict

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

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

# print(combine\_dicts(d1, d2)) # Output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})

1. Write a Python program to find the highest 3 values in a dictionary.

def find\_highest\_3\_values(dict1):

"""

Finds the highest 3 values in a dictionary.

Args:

dict1 (dict): The dictionary to search.

Returns:

A list of the highest 3 values in the dictionary.

"""

# Convert the dictionary to a list of tuples.

list1 = list(dict1.items())

# Sort the list by the values.

list1.sort(key=lambda x: x[1], reverse=True)

# Return the highest 3 values.

return list1[:3]

dict1 = {'a': 1, 'b': 3, 'c': 2, 'd': 5, 'e': 4}

print(find\_highest\_3\_values(dict1)) # Output: [('d', 5), ('b', 3), ('e', 4)]

# Write a Python program to match key values in two dictionaries. Sample dictionary: {'key1': 1, 'key2': 3, 'key3': 2}, {'key1': 1, 'key2': 2} Expected output: key1: 1 is present in both x and y

def match\_key\_values(x, y):

"""

Matches the key-values in two dictionaries.

Args:

x (dict): The first dictionary.

y (dict): The second dictionary.

Returns:

A list of key-value pairs that are present in both dictionaries.

"""

matches = []

for key, value in x.items():

if key in y and y[key] == value:

matches.append((key, value))

return matches

x = {'key1': 1, 'key2': 3, 'key3': 2}

y = {'key1': 1, 'key2': 2}

print(match\_key\_values(x, y)) # Output: [('key1', 1)]

1. Write a Python program to check if all dictionaries in a list are empty or not. Sample list : [{},{},{}]

# Return value : True Sample list : [{1,2},{},{}] Return value : False

# def check\_empty\_dicts(list\_of\_dicts):

# """

# Checks if all dictionaries in a list are empty.

# Args:

# list\_of\_dicts: A list of dictionaries.

# Returns:

# True if all dictionaries are empty, False otherwise.

# """

# for d in list\_of\_dicts:

# if d:

# return False

# return True

# list1 = [{},{},{}]

# list2 = [{1,2},{},{}]

# print(check\_empty\_dicts(list1)) # True

# print(check\_empty\_dicts(list2)) # False

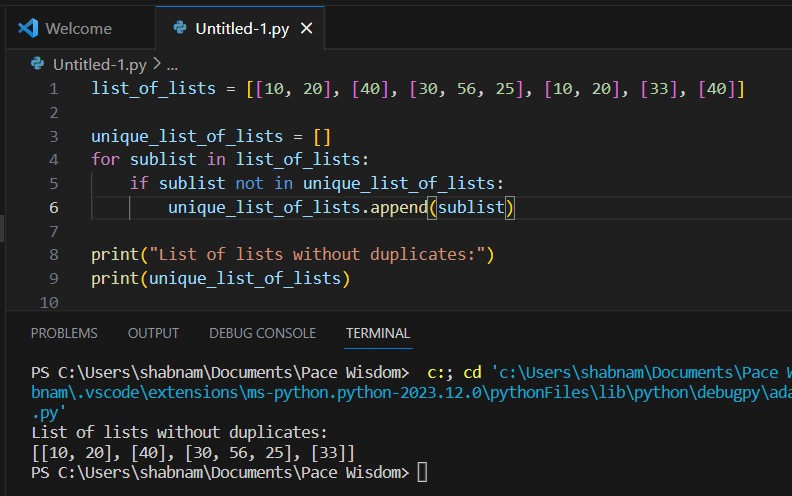
1. Write a Python program to remove duplicates from a list of lists. Sample list : [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]

# New List : [[10, 20], [30, 56, 25], [33], [40]] ANS:

list\_of\_lists = [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]

unique\_list\_of\_lists = [] for sublist in list\_of\_lists:

if sublist not in unique\_list\_of\_lists: unique\_list\_of\_lists.append(sublist)

print("List of lists without duplicates:") print(unique\_list\_of\_lists)

# Write a Python program to extend a list without append. Sample data: [10, 20, 30]

[40, 50, 60]

# Expected output : [40, 50, 60, 10, 20, 30]

ANS:

list1 = [10, 20, 30]

list2 = [40, 50, 60]

list1 += list2 print("Extended list:", list1)

