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

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)

**2.** 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}

d = {0: 10, 1: 20}

d[2] = 30

print(d)

**3.** 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}

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)

**4.** Write a Python script to check if a given key already exists in a dictionary.

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

key\_to\_check = 1

if key\_to\_check in sample\_dict:

print("Key exists")

else:

print("Key does not exist")

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

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))

**6.** 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}

n = 5

square\_dict = {x: x\*x for x in range(1, n+1)}

print(square\_dict)

**7.** Write a Python script to merge two Python dictionaries.

d1 = {'a': 1, 'b': 2}

d2 = {'c': 3, 'd': 4}

d1.update(d2)

print(d1)

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

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

total = sum(sample\_dict.values())

print(total)

**9.** Write a Python program to multiply all the items in a dictionary.

sample\_dict = {1: 2, 2: 3, 3: 4}

result = 1

for value in sample\_dict.values():

result \*= value

print(result)

**10.** Write a Python program to remove a key from a dictionary.

d = {'a': 1, 'b': 2, 'c': 3}

key = 'b'

if key in d:

del d[key]

print(d)

OR

sample\_dict = {0: 10, 1: 20, 2: 30}

key\_to\_remove = 1

sample\_dict.pop(key\_to\_remove, None)

print(sample\_dict)

**11.** Write a Python program to sort a dictionary by key.

d = {'b': 1, 'a': 2, 'c': 3}

print(dict(sorted(d.items())))

**12.** Write a Python program to get the maximum and minimum value in a dictionary.

d = {'a': 1, 'b': 2, 'c': 3}

print(max(d.values()), min(d.values()))

**13.** Write a Python program to remove duplicates from Dictionary.

d = {'a': 1, 'b': 1, 'c': 2}

print(dict(d))

OR

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

unique\_dict = {key: value for key, value in sample\_dict.items()}

print(unique\_dict)

**14.** Write a Python program to check a dictionary is empty or not.

empty\_dict = {}

if not bool(empty\_dict):

print("Dictionary is empty")

else:

print("Dictionary is not empty")

**15.** 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}

combined\_dict = Counter(d1) + Counter(d2)

print(combined\_dict)

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

sample\_dict = {'a': 10, 'b': 50, 'c': 20, 'd': 40, 'e': 30}

highest\_values = sorted(sample\_dict.values(), reverse=True)[:3]

print(highest\_values)

**17.** 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

dict\_x = {'key1': 1, 'key2': 3, 'key3': 2}

dict\_y = {'key1': 1, 'key2': 2}

for key in dict\_x.keys():

if key in dict\_y and dict\_x[key] == dict\_y[key]:

print(f"{key}: {dict\_x[key]} is present in both x and y")

**18.** 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

dicts1 = [{}, {}, {}]

dicts2 = [{1, 2}, {}, {}]

all\_empty1 = all(not bool(d) for d in dicts1)

all\_empty2 = all(not bool(d) for d in dicts2)

print(all\_empty1) # True

print(all\_empty2) # False

**19.** 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]]

sample\_list = [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]

new\_list = [list(item) for item in set(tuple(sublist) for sublist in sample\_list)]

print(new\_list)

**20.** 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]

sample\_data = [10, 20, 30]

extension\_data = [40, 50, 60]

extended\_list = extension\_data + sample\_data

print(extended\_list)

OR

list1 = [10, 20, 30]

list2 = [40, 50, 60]

list1[:0] = list2

print(list1)