ASSIGNMENT 7(List and Dictionary)

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

```
my_dict = {'apple': 3, 'banana': 1, 'orange': 2}
sorted_dict_asc = dict(sorted(my_dict.items(), key=lambda x: x[1]))
sorted_dict_desc = dict(sorted(my_dict.items(), key=lambda x: x[1], reverse=True))
print("Original Dictionary:", my_dict)
print("Ascending Order:", sorted_dict_asc)
print("Descending Order:", sorted_dict_desc)
```

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}

```
sample_dict = {0: 10, 1: 20}
sample_dict[2] = 30
print(sample_dict)
```

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 = {}
for d in (dic1, dic2, dic3):
result.update(d)
print(result)
```

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

```
my_dict = {'apple': 5, 'banana': 10, 'orange': 15}
def key_exists(key, dictionary):
if key in dictionary:
return True
else:
return False
print(key_exists('apple', my_dict))  # Output: True
print(key_exists('grape', my_dict))  # Output: False
```

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

```
student_scores = {
'Alice': 85,
'Bob': 92,
'Charlie': 78,
'David': 80
}
print("Iterating over keys:")
for key in student_scores:
print(key)
print("Iterating over values:")
for value in student_scores.values():
print(value)
print("Iterating over items:")
for key, value in student_scores.items():
print(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
squares_dict = {x: x*x for x in range(1, n+1)}
print(squares_dict)
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

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

```
dict1 = {"a": 1, "b": 2}
dict2 = {"c": 3, "d": 4}
merged_dict = {**dict1, **dict2}
print(merged_dict)
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

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

```
my_dict = {"a": 10, "b": 20, "c": 30}
sum_of_values = sum(my_dict.values())
print(sum_of_values)
```

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

```
my_dict = {"a": 10, "b": 20, "c": 30}
from functools import reduce
product_of_values = reduce(lambda x, y: x * y, my_dict.values())
print(product_of_values)
```

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

```
my_dict = {"a": 10, "b": 20, "c": 30}
my_dict.pop('b')
print(my_dict)
{'a': 10, 'c': 30}
```

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

```
my_dict = {"c": 30, "a": 10, "b": 20}
sorted_dict = dict(sorted(my_dict.items()))
print(sorted_dict)
{'a': 10, 'b': 20, 'c': 30}
```

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

```
my_dict = {"a": 10, "b": 20, "c": 30}
max_value = max(my_dict.values())
min_value = min(my_dict.values())
print("Maximum value:", max_value)
print("Minimum value:", min_value)
```

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

```
my_dict = {"a": 10, "b": 20, "c": 10, "d": 30, "e": 20}
new_dict = {}
for key, value in my_dict.items():
if value not in new_dict.values():
new_dict[key] = value
print(new_dict)
```

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

```
my_dict = {}
if not bool(my_dict):
print("The dictionary is empty")
else:
print("The 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})

```
d1 = {'a': 100, 'b': 200, 'c': 300}
d2 = {'a': 300, 'b': 200, 'd': 400}
counter = Counter(d1) + Counter(d2)
result_dict = dict(counter)
print(result_dict)
```

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

```
my_dict = {"a": 10, "b": 50, "c": 30, "d": 40, "e": 20}
sorted_dict = sorted(my_dict.items(), key=lambda x: x[1], reverse=True)
for key, value in sorted_dict[:3]:
print(key, value)
```

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

```
dict1 = {'key1': 1, 'key2': 3, 'key3': 2}
dict2 = {'key1': 1, 'key2': 2}
for key in dict1.keys():
if key in dict2:
if dict1[key] == dict2[key]:
print(f"{key}: {dict1[key]} is present in both dict1 and dict2")
```

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

```
def check_empty_dicts(lst):
"""
This function takes a list of dictionaries as input and returns True if all the dictionaries
in the list are empty, and False otherwise.
"""
for d in lst:
if bool(d):
return False
return True
lst1 = [{},{},{}]
lst2 = [{1,2},{},{}]
print(check_empty_dicts(lst1)) # Output: True
print(check_empty_dicts(lst2)) # Output: 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]]

```
def remove_duplicates(lst):
"""
This function takes a list of lists as input and returns a new list with duplicates removed.
"""
new_lst = []
for sublist in lst:
if sublist not in new_lst:
new_lst.append(sublist)
return new_lst
lst = [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]
new_lst = remove_duplicates(lst)
print(new_lst) # Output: [[10, 20], [40], [30, 56, 25], [33]]
```

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]

```
def extend_list(lst1, lst2):
"""
This function takes two lists as input and returns a new list which is the extension of the second list
followed by the first list.
"""
new_lst = []
for item in lst2:
new_lst.insert(0, item)
for item in lst1:
new_lst.insert(0, item)
return new_lst
lst1 = [10, 20, 30]
lst2 = [40, 50, 60]
new_lst = extend_list(lst1, lst2)
print(new_lst) # Output: [40, 50, 60, 10, 20, 30]
```