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

In [ ]:
```
_1 = {1:33,22:32,4:99,0:1}
_1_ans = _1.items() # this give list with(key, value) tuple
_1_ascending = sorted(_1_ans,key=lambda x:x[1])
_1_decent = sorted(_1_ans,key=lambda x:x[1])[::-1]
print(_1_ascending)
print(_1_decent)
```

```
[(0, 1), (22, 32), (1, 33), (4, 99)]
[(4, 99), (1, 33), (22, 32), (0, 1)]
```

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}

In [ ]:
```
_2 = {0: 10, 1: 20}
_2[2] = 30
print(_2)
```

```
{0: 10, 1: 20, 2: 30}
```

3. Write a Python script to concatenate the 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}

In [ ]:
```
_3_dic1={1:10, 2:20}
_3_dic2={3:30, 4:40}
_3_dic3={5:50,6:60}
_3_ans = {}
_3_ans.update(_3_dic1)
_3_ans.update(_3_dic2)
_3_ans.update(_3_dic3)
print(_3_ans)
```

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

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

In [ ]:
```
_4_dict = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
Key = 1
if  Key in _4_dict.keys():
    print("yes")
else:
    print("no")
```

```
yes
```

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

In [ ]:
```
_5_dict = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
for (k,v) in _5_dict.items():
```

```
    print(k,v)
```

```
1 10
2 20
3 30
4 40
5 50
6 60
```

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}

In [ ]:
```
_6_dict = {}
n = 11
for i in range(1,n+1):
    _6_dict[i] = i*i
print(_6_dict)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121}
```

7. Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are the square of the keys. Sample Dictionary {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}

In [ ]:
```
_7_dict = {}
for i in range(1,16):
    _7_dict[i] = i*i
print(_7_dict)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 1
2: 144, 13: 169, 14: 196, 15: 225}
```

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

In [ ]:
```
_8_dic1={1:10, 2:20}
_8_dic2={3:30, 4:40}
_8_ans = {} # merged dict
_8_ans.update(_8_dic1)
_8_ans.update(_8_dic2)
print(_3_ans)
```

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

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

In [ ]:
```
_9_dict = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
for (k,v) in _9_dict.items():
    print(k,v)
```

```
1 10
2 20
3 30
4 40
5 50
6 60
```

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

In [ ]:
```
_10_dict = {1: 10, 2: 20}
_10_sum = 0
for (k,v) in _10_dict.items():
    _10_sum += k+v
print(_10_sum)
```

```
33
```

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

In [ ]:
```
_11_dict = {1: 10, 2: 20}
_11_product = 1
for (k,v) in _11_dict.items():
    _11_product *= k*v
print(_11_product)
```

```
400
```

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

In [ ]:
```
_12_dict = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
del(_12_dict[1])
print(_12_dict)
```

```
{2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

13. Write a Python program to map two lists into a dictionary.

In [ ]:
```
_13_k = [1, 2, 3, 4, 5, 6]
_13_v = [10, 20, 30, 40, 50, 60]
_13_dict = {}
for i in range(len(_13_k)):
    _13_dict[_13_k[i]] = _13_v[i]
print(_13_dict)
```

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

14. Write a Python program to sort a given dictionary by key.

In [ ]:
```
_14 = {1:33,22:32,4:99,0:1}
_14_ans = _14.items() # this give list with(key, value) tuple
_14_ascending = sorted(_14_ans,key=lambda x:x[0])
_14_decent = sorted(_14_ans,key=lambda x:x[0])[::-1]
print(_14_ascending)
print(_14_decent)
```

```
[(0, 1), (1, 33), (4, 99), (22, 32)]
[(22, 32), (4, 99), (1, 33), (0, 1)]
```

15. Write a Python program to get the maximum and minimum values of a dictionary.

In [ ]:
```
_15_dict = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
mx = max(_15_dict.values())#max
mi = min(_15_dict.values())#min
print(mx,mi)
```

```
60 10
```

16. Write a Python program to get a dictionary from an object's fields.

In [ ]:
```
class AI:
    def __init__(self):
        self.a = "thus"
        self.b = 9

print(AI.__dict__)
```

```
{'__module__': '__main__', '__init__': <function AI.__init__ at 0x0000018AF661A66
0>, '__dict__': <attribute '__dict__' of 'AI' objects>, '__weakref__': <attribute
'__weakref__' of 'AI' objects>, '__doc__': None}
```

17. Write a Python program to remove duplicates from the dictionary.

In [ ]:
```
_17_ans = dict()
_17_dict = {1: 10, 2: 20, 3: 30,3: 30, 4: 40, 5: 50, 6: 60, 6: 60}
for key, val in _17_dict.items():
        if val not in _17_ans.values():
                _17_ans[key] = val
print(_17_ans)
```

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

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

In [ ]:
```
_18_dict_1 = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
_18_dict_2 = {}

print(_18_dict_1 == {})
print(_18_dict_2 == {})
```

```
False
True
```

19. Write a Python program to combine two dictionary by 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})

In [ ]:
```
d1 = {'a': 100, 'b': 200, 'c': 300}
d2 = {'a': 300, 'b': 200, 'd': 400}

combined_dict = {}
```

```
for key in set(d1.keys()) | set(d2.keys()):
    combined_dict[key] = d1.get(key, 0) + d2.get(key, 0)

print(combined_dict)
```

```
{'b': 400, 'd': 400, 'c': 300, 'a': 400}
```

20. Write a Python program to print all distinct values in a dictionary. Sample Data : [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S005"}, {"V":"S009"}, {"VIII":"S007"}] Expected Output : Unique Values: {'S005', 'S002', 'S007', 'S001', 'S009'}

In [ ]:
```
data = [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S00

distinct_values = set()

for item in data:
    for value in item.values():
        distinct_values.add(value)

print("Unique Values:", distinct_values)
```

```
Unique Values: {'S001', 'S002', 'S007', 'S005', 'S009'}
```

21. Write a Python program to create and display all combinations of letters, selecting each letter from a different key in a dictionary. Sample data : {'1':['a','b'], '2':['c','d']} Expected Output: ac ad bc bd

In [ ]:
```
def generate_combinations(dictionary):
    keys = dictionary.keys()
    combinations = [[]]

    for key in keys:
        new_combinations = []
        for letter in dictionary[key]:
            for combination in combinations:
                new_combinations.append(combination + [letter])
        combinations = new_combinations

    for combination in combinations:
        print(''.join(combination))


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

generate_combinations(my_dict)
```

```
ac
bc
ad
bd

bc
ad
bd
```

22. Write a Python program to find the highest 3 values of corresponding keys in a dictionary.

In [ ]:
```
my_dict = {
    'key1': 30,
    'key2': 15,
    'key3': 45,
    'key4': 20,
    'key5': 50
}

highest_values = []
highest_keys = []

for key, value in my_dict.items():
    if len(highest_values) < 3:
        highest_values.append(value)
        highest_keys.append(key)
    else:
        min_index = highest_values.index(min(highest_values))
        if value > highest_values[min_index]:
            highest_values[min_index] = value
            highest_keys[min_index] = key

for key, value in zip(highest_keys, highest_values):
    print(f"{key}: {value}")
```

```
key1: 30
key5: 50
key3: 45
```

23. Write a Python program to combine values in a list of dictionaries. Sample data: [{'item': 'item1', 'amount': 400}, {'item': 'item2', 'amount': 300}, {'item': 'item1', 'amount': 750}] Expected Output: Counter({'item1': 1150, 'item2': 300})

In [ ]:
```
data = [{'item': 'item1', 'amount': 400},
        {'item': 'item2', 'amount': 300},
        {'item': 'item1', 'amount': 750}]

combined_data = {}

for d in data:
    item = d['item']
    amount = d['amount']
    if item in combined_data:
        combined_data[item] += amount
    else:
        combined_data[item] = amount

print(combined_data)
```

```
{'item1': 1150, 'item2': 300}
```

24. Write a Python program to create a dictionary from a string. Note: Track the count of the letters from the string. Sample string : 'w3resource' Expected output: {'w': 1, '3':

1, 'r': 2, 'e': 2, 's': 1, 'o': 1, 'u': 1, 'c': 1}

In [ ]:
```
my_string = 'w3resource'
letter_counts = {}

for letter in my_string:
    if letter in letter_counts:
        letter_counts[letter] += 1
    else:
        letter_counts[letter] = 1

print(letter_counts)
```

```
{'w': 1, '3': 1, 'r': 2, 'e': 2, 's': 1, 'o': 1, 'u': 1, 'c': 1}
```

25. Write a Python program to print a dictionary in table format.

In [ ]:
```
my_dict = {
    'name': 'John',
    'age': 30,
    'city': 'New York',
    'occupation': 'Engineer'
}

max_key_length = max(len(key) for key in my_dict)
max_value_length = max(len(str(value)) for value in my_dict.values())

print(f"+{'-' * (max_key_length + 2)}+{'-' * (max_value_length + 2)}+")
for key, value in my_dict.items():
    print(f"| {key:<{max_key_length}} | {value:<{max_value_length}} |")
print(f"+{'-' * (max_key_length + 2)}+{'-' * (max_value_length + 2)}+")
```

```
+------------+----------+
| name       | John     |
| age        | 30       |
| city       | New York |
| occupation | Engineer |
+------------+----------+
```

26. Write a Python program to count the values associated with a key in a dictionary.
Expected Output: 6 2

In [ ]:
```
my_dict = {
    'key1': 6,
    'key2': 2,
    'key3': 6,
    'key4': 3,
    'key5': 6
}

value_counts = {}

for key, value in my_dict.items():
    if value in value_counts:
        value_counts[value] += 1
    else:
        value_counts[value] = 1
```

```
for value, count in value_counts.items():
    print(f"{value}: {count}")
```

```
6: 3
2: 1
3: 1
```

27. Write a Python program to convert a list into a nested dictionary of keys.

In [ ]:
```
my_list = ['a', 'b', 'c', 'd']
nested_dict = {}
current_dict = nested_dict
for item in my_list[:-1]:
    current_dict[item] = {}
    current_dict = current_dict[item]

current_dict[my_list[-1]] = None

print(nested_dict)
```

```
{'a': {'b': {'c': {'d': None}}}}
```

28. Write a Python program to sort a list alphabetically in a dictionary.

In [ ]:
```
my_dict = {
    'fruits': ['apple', 'orange', 'banana', 'grape'],
    'colors': ['red', 'blue', 'green', 'yellow'],
    'animals': ['dog', 'cat', 'elephant', 'lion']
}

for key, value in my_dict.items():
    my_dict[key] = sorted(value)

print(my_dict)
```

```
{'fruits': ['apple', 'banana', 'grape', 'orange'], 'colors': ['blue', 'green', 'r
ed', 'yellow'], 'animals': ['cat', 'dog', 'elephant', 'lion']}
```

29. Write a Python program to remove spaces from dictionary keys.

In [ ]:
```
my_dict = {
    'key 1': 1,
    'key 2': 2,
    'key 3': 3,
    'key 4': 4,
    'key 5': 5
}

new_dict = {key.replace(' ', ''): value for key, value in my_dict.items()}

print(new_dict)
```

```
{'key1': 1, 'key2': 2, 'key3': 3, 'key4': 4, 'key5': 5}
```

30. Write a Python program to get the top three items in a shop. Sample data: {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24} Expected Output: item4 55

item1 45.5 item3 41.3

In [ ]:
```
shop_items = {
    'item1': 45.50,
    'item2': 35,
    'item3': 41.30,
    'item4': 55,
    'item5': 24
}

first_item = None
second_item = None
third_item = None

for item, price in shop_items.items():
    if first_item is None or price > shop_items[first_item]:
        third_item = second_item
        second_item = first_item
        first_item = item
    elif second_item is None or price > shop_items[second_item]:
        third_item = second_item
        second_item = item
    elif third_item is None or price > shop_items[third_item]:
        third_item = item

print(first_item, shop_items[first_item])
print(second_item, shop_items[second_item])
print(third_item, shop_items[third_item])
```

```
item4 55
item1 45.5
item3 41.3
```

31. Write a Python program to get the key, value and item in a dictionary.

In [ ]:
```
my_dict = {
    'key1': 1,
    'key2': 2,
    'key3': 3,
    'key4': 4,
    'key5': 5
}

for key, value in my_dict.items():
    print("Key:", key)
    print("Value:", value)
    print("Item:", (key, value))
    print("---")
```

```
Key: key1
Value: 1
Item: ('key1', 1)
---
Key: key2
Value: 2
Item: ('key2', 2)
---
Key: key3
Value: 3
Item: ('key3', 3)
---
Key: key4
Value: 4
Item: ('key4', 4)
---
Key: key5
Value: 5
Item: ('key5', 5)
---
```

32. Write a Python program to print a dictionary line by line.

In [ ]:
```python
my_dict = {
    'key1': 1,
    'key2': 2,
    'key3': 3,
    'key4': 4,
    'key5': 5
}

for key, value in my_dict.items():
    print(f"{key}: {value}")
```

```
key1: 1
key2: 2
key3: 3
key4: 4
key5: 5
```

33. Write a Python program to check if multiple keys exist in a dictionary.

In [ ]:
```python
my_dict = {
    'key1': 1,
    'key2': 2,
    'key3': 3,
    'key4': 4,
    'key5': 5
}

keys_to_check = ['key1', 'key3', 'key6']

for key in keys_to_check:
    if key in my_dict:
        print(f"The key '{key}' exists in the dictionary.")
    else:
        print(f"The key '{key}' does not exist in the dictionary.")
```

```
The key 'key1' exists in the dictionary.
The key 'key3' exists in the dictionary.
The key 'key6' does not exist in the dictionary.
```

34. Write a Python program to count the number of items in a dictionary value that is a list.

In [ ]:
```python
my_dict = {
    'list1': [1, 2, 3],
    'list2': [4, 5],
    'list3': [6, 7, 8, 9],
    'non_list': 'Not a list',
    'empty_list': []
}

list_count = 0

for value in my_dict.values():
    if type(value) is list:
        list_count += 1

print("Number of items that are lists:", list_count)
```

```
Number of items that are lists: 4
```

35. Write a Python program to sort Counter by value. Sample data : {'Math':81, 'Physics':83, 'Chemistry':87} Expected data: [('Chemistry', 87), ('Physics', 83), ('Math', 81)]

In [ ]:
```python
from collections import Counter

def sort_counter_by_value(counter_dict):
    sorted_items = sorted(counter_dict.items(), key=lambda x: x[1], reverse=True
    return sorted_items


sample_data = {'Math': 81, 'Physics': 83, 'Chemistry': 87}


sorted_data = sort_counter_by_value(sample_data)


print(sorted_data)
```

```
[('Chemistry', 87), ('Physics', 83), ('Math', 81)]
```

36. Write a Python program to create a dictionary from two lists without losing duplicate values. Sample lists: ['Class-V', 'Class-VI', 'Class-VII', 'Class-VIII'], [1, 2, 2, 3] Expected Output: defaultdict(<class 'set'>, {'Class-V': {1}, 'Class-VI': {2}, 'Class- VII': {2}, 'Class-VIII': {3}})

In [ ]:
```python
class_names = ['Class-V', 'Class-VI', 'Class-VII', 'Class-VIII']
class_numbers = [1, 2, 2, 3]

result_dict = {}
```

```
for i in range(len(class_names)):
    class_name = class_names[i]
    class_number = class_numbers[i]

    if class_name in result_dict:
        result_dict[class_name].add(class_number)
    else:
        result_dict[class_name] = {class_number}

print(result_dict)
```

```
{'Class-V': {1}, 'Class-VI': {2}, 'Class-VII': {2}, 'Class-VIII': {3}}
```

37. Write a Python program to replace dictionary values with their sums.

In [ ]:
```
#37
my_dict = {'key1': 1, 'key2': 3, 'key3': 2}

sum_values = sum(my_dict.values())

for key in my_dict:
    my_dict[key] = sum_values

print(my_dict)
```

```
{'key1': 6, 'key2': 6, 'key3': 6}
```

38. 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

In [ ]:
```
# 38
x = {'key1': 1, 'key2': 3, 'key3': 2}
y = {'key1': 1, 'key2': 3}

for key_x, value_x in x.items():
    for key_y, value_y in y.items():
        if key_x == key_y and value_x == value_y:
            print(f"{key_x}: {value_x} is present in both x and y")
```

```
key1: 1 is present in both x and y
key2: 3 is present in both x and y
```