1. Write a Python program to Extract Unique values dictionary values?
2. Write a Python program to find the sum of all items in a dictionary?
3. Write a Python program to Merging two Dictionaries?
4. Write a Python program to convert key-values list to flat dictionary?
5. Write a Python program to insertion at the beginning in OrderedDict?
6. Write a Python program to check order of character in string using OrderedDict()?
7. Write a Python program to sort Python Dictionaries by Key or Value?

Ans: Sure! Here are the Python programs to solve each of the problems you mentioned:

1. Extract Unique values from dictionary values:

```python

def extract\_unique\_values(dictionary):

unique\_values = set()

for value in dictionary.values():

if isinstance(value, list):

unique\_values.update(value)

else:

unique\_values.add(value)

return list(unique\_values)

# Example usage:

my\_dict = {'a': [1, 2, 3], 'b': [2, 3, 4], 'c': [3, 4, 5]}

unique\_values = extract\_unique\_values(my\_dict)

print(unique\_values)

```

2. Find the sum of all items in a dictionary:

```python

def sum\_dictionary\_items(dictionary):

total\_sum = sum(dictionary.values())

return total\_sum

# Example usage:

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

sum\_of\_items = sum\_dictionary\_items(my\_dict)

print(sum\_of\_items)

```

3. Merging two dictionaries:

```python

def merge\_dictionaries(dict1, dict2):

merged\_dict = {\*\*dict1, \*\*dict2}

return merged\_dict

# Example usage:

dict1 = {'a': 1, 'b': 2}

dict2 = {'c': 3, 'd': 4}

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

print(merged\_dict)

```

4. Convert key-values list to a flat dictionary:

```python

def convert\_to\_flat\_dictionary(key\_value\_list):

flat\_dict = {}

for key, value in key\_value\_list:

flat\_dict[key] = value

return flat\_dict

# Example usage:

key\_value\_list = [('a', 1), ('b', 2), ('c', 3)]

flat\_dict = convert\_to\_flat\_dictionary(key\_value\_list)

print(flat\_dict)

```

5. Insertion at the beginning in OrderedDict:

```python

from collections import OrderedDict

def insert\_at\_beginning(dictionary, key, value):

dictionary.move\_to\_end(key, last=False)

dictionary[key] = value

# Example usage:

my\_dict = OrderedDict([('a', 1), ('b', 2), ('c', 3)])

insert\_at\_beginning(my\_dict, 'x', 10)

print(my\_dict)

```

6. Check the order of characters in a string using OrderedDict:

```python

from collections import OrderedDict

def check\_order\_of\_characters(string, pattern):

ordered\_dict = OrderedDict.fromkeys(string)

pattern\_length = 0

for key in ordered\_dict:

if key == pattern[pattern\_length]:

pattern\_length += 1

if pattern\_length == len(pattern):

return True

return False

# Example usage:

my\_string = 'Hello World'

my\_pattern = 'Wo'

is\_ordered = check\_order\_of\_characters(my\_string, my\_pattern)

print(is\_ordered)

```

7. Sort Python Dictionaries by Key or Value:

To sort dictionaries by key:

```python

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

sorted\_dict = dict(sorted(my\_dict.items()))

print(sorted\_dict)

```

To sort dictionaries by value:

```python

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

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

print(sorted\_dict)

```

I hope these programs help you! Let me know if you have any further questions.