1. Write a Python program to Extract Unique values dictionary values?

In [2]:
```python
def extract_unique_values(d):
    # Create an empty set to store the unique values
    unique_values = set()

    # Iterate over the dictionary values
    for value in d.values():
        # If the value is not already in the set, add it
        if value not in unique_values:
            unique_values.add(value)

    # Return the set of unique values
    return unique_values
```

2. Write a Python program to find the sum of all items in a dictionary?

In [3]:
```python
my_dict = {'apple': 5, 'banana': 7, 'orange': 3}

# get a list of all the values in the dictionary
values = my_dict.values()

# find the sum of the values
sum_of_values = sum(values)

print("The sum of all items in the dictionary is:", sum_of_values)
```

```
The sum of all items in the dictionary is: 15
```

3. Write a Python program to Merging two Dictionaries?

In [4]:
```python
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}
dict1.update(dict2)
print(dict1) # Output: {'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

```
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

4. Write a Python program to convert key-values list to flat dictionary?

In [5]:
```python
def flatten_dict(lst):
    """
    Function to convert key-values list to flat dictionary
    """
    flat_dict = {}
    for pair in lst:
        if isinstance(pair[1], dict):
            # recursively flatten sub-dictionaries
            sub_dict = flatten_dict(pair[1].items())
            for sub_key, sub_value in sub_dict.items():
                flat_dict[f"{pair[0]}.{sub_key}"] = sub_value
        else:
            flat_dict[pair[0]] = pair[1]
    return flat_dict

# example usage
lst = [('a', 1), ('b', {'x': 2, 'y': 3}), ('c', 4)]
flat_dict = flatten_dict(lst)
print(flat_dict)
# Output: {'a': 1, 'b.x': 2, 'b.y': 3, 'c': 4}
```

```
{'a': 1, 'b.x': 2, 'b.y': 3, 'c': 4}
```

5. Write a Python program to insertion at the beginning in OrderedDict?

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

# Creating an empty OrderedDict
my_dict = OrderedDict()
```

```
# Inserting elements into the OrderedDict
my_dict['b'] = 2
my_dict['c'] = 3

# Inserting element at the beginning of the OrderedDict
my_dict.insert(0, 'a', 1)

# Displaying the OrderedDict
print(my_dict)
```

6. Write a Python program to check order of character in string using OrderedDict()?

In [9]:
```
from collections import OrderedDict

def check_order(input_str, pattern):
    # creating an OrderedDict to store the count of each character
    # in the input string
    dict = OrderedDict.fromkeys(input_str, 0)

    # loop through each character in the input string
    for char in input_str:
        dict[char] += 1

    # create an empty string to store the ordered characters
    ordered_str = ""

    # loop through each character in the pattern string
    for char in pattern:
        if char in dict:
            # add the character to the ordered string
            ordered_str += char
            # remove the character from the dict to avoid duplicates
            dict.pop(char)

    # return True if the ordered string matches the pattern string,
    # otherwise return False
    if ordered_str == pattern:
        return True
    else:
        return False

# example usage
input_str = "hello world"
pattern = "hlo"
if check_order(input_str, pattern):
    print("The characters in pattern are in order in the input string.")
else:
    print("The characters in pattern are not in order in the input string.")
```

```
The characters in pattern are in order in the input string.
```

7. Write a Python program to sort Python Dictionaries by Key or Value?

In [10]:
```
# A sample dictionary
my_dict = {"apple": 3, "banana": 2, "orange": 1}

# Sorting by key
sorted_dict_by_key = dict(sorted(my_dict.items()))

print("Dictionary sorted by key:", sorted_dict_by_key)

# Sorting by value
sorted_dict_by_value = dict(sorted(my_dict.items(), key=lambda x: x[1]))

print("Dictionary sorted by value:", sorted_dict_by_value)
```

```
Dictionary sorted by key: {'apple': 3, 'banana': 2, 'orange': 1}
Dictionary sorted by value: {'orange': 1, 'banana': 2, 'apple': 3}
```

In [ ]: