In [1]:
```
# Python3 Program to find sum of
# all items in a Dictionary

# Function to print sum


def returnSum(myDict):

    list = []
    for i in myDict:
        list.append(myDict[i])
    final = sum(list)

    return final


# Driver Function
dict = {'a': 100, 'b': 200, 'c': 300}
print("Sum :", returnSum(dict))
```

```
Sum : 600
```

In [2]:
```
dict={'a':1,'b':22,'c':333}
list=[]
for i in dict:
    list.append(dict[i])
a=sum(list)
print(a)
```

```
356
```

In [4]:
```
def add(arr):
    lst=[]
    for i in arr:
        lst.append(arr[i])
    final=sum(lst)
    print(final)
arr={'a': 100, 'b': 200, 'c': 300}
add(arr)
```

```
600
```

In [12]:
```
dict={'Manu':22,'krishnq':22,'camel':333}
b=dict.pop('Manu')
print(dict)
value=dict.pop('camel','no key found')
print(value)
```

```
{'krishnq': 22, 'camel': 333}
333
```

In [11]:
```
# Initializing dictionary
test_dict = {"Arushi": 22, "Anuradha": 21, "Mani": 21, "Haritha": 21}

# Printing dictionary before removal
print("The dictionary before performing remove is : " + str(test_dict))

# Using pop() to remove a dict. pair
# removes Mani
removed_value = test_dict.pop('Mani')

# Printing dictionary after removal
print("The dictionary after remove is : " + str(test_dict))
print("The removed key's value is : " + str(removed_value))

print('\r')

# Using pop() to remove a dict. pair
# doesn't raise exception
# assigns 'No Key found' to removed_value
removed_value = test_dict.pop('Manjeet', 'No Key found')

# Printing dictionary after removal
print("The dictionary after remove is : " + str(test_dict))
print("The removed key's value is : " + str(removed_value))
```

```
The dictionary before performing remove is : {'Arushi': 22, 'Anuradha': 21,
'Mani': 21, 'Haritha': 21}
The dictionary after remove is : {'Arushi': 22, 'Anuradha': 21, 'Haritha': 2
1}
The removed key's value is : 21

The dictionary after remove is : {'Arushi': 22, 'Anuradha': 21, 'Haritha': 2
1}
The removed key's value is : No Key found
```

In [20]:
```
list_of_dicts = [{'name': 'Alice', 'age': 30},
                 {'name': 'Dob', 'age': 25},
                 {'name': 'Charlie', 'age': 35}]

# Define a custom sorting function
def sort_by_age(d):
    return d['name']

# Sort the list using the custom function
sorted_list = sorted(list_of_dicts, key=sort_by_age)

print(sorted_list)
```

```
[{'name': 'Alice', 'age': 30}, {'name': 'Charlie', 'age': 35}, {'name': 'Do
b', 'age': 25}]
```

In [23]:
```
list_of_dicts = [{'name': 'Alice', 'age': 30},
                 {'name': 'Bob', 'age': 25},
                 {'name': 'Charlie', 'age': 35}]
def sort(a):
    return a['name']
sorted_list=sorted(list_of_dicts,key=sort)
print(sorted_list)
```

```
[{'name': 'Alice', 'age': 30}, {'name': 'Bob', 'age': 25}, {'name': 'Charli
e', 'age': 35}]
```

In [26]:
```
dict1 = {'a': 1, 'b': 2}
dict2 = {'d': 3, 'c': 4}

merged_dict = {**dict1, **dict2}

print(merged_dict)
```

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

In [27]:
```
from collections import OrderedDict
original_dict = OrderedDict([('a', 1), ('b', 2), ('c', 3)])
new_key = 'x'
new_value = 100
new_dict = OrderedDict([(new_key, new_value)])
new_dict.update(original_dict)
print(new_dict)
```

```
OrderedDict([('x', 100), ('a', 1), ('b', 2), ('c', 3)])
```

In [28]:
```
my_dict = {'b': 3, 'a': 2, 'c': 1}

sorted_dict_by_keys = dict(sorted(my_dict.items()))

print(sorted_dict_by_keys)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4408\306465876.py in <module>
      1 my_dict = {'b': 3, 'a': 2, 'c': 1}
      2
----> 3 sorted_dict_by_keys = dict(sorted(my_dict.items()))
      4
      5 print(sorted_dict_by_keys)

TypeError: 'dict' object is not callable
```

In [ ]: