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

**ANS:-**

**# Python3 code to demonstrate working of**

**# Extract Unique values dictionary values**

**# Using set comprehension + values() + sorted()**

**# initializing dictionary**

**test\_dict = {'gfg': [5, 6, 7, 8],**

**'is': [10, 11, 7, 5],**

**'best': [6, 12, 10, 8],**

**'for': [1, 2, 5]}**

**# printing original dictionary**

**print("The original dictionary is : " + str(test\_dict))**

**# Extract Unique values dictionary values**

**# Using set comprehension + values() + sorted()**

**res = list(sorted({ele for val in test\_dict.values() for ele in val}))**

**# printing result**

**print("The unique values list is : " + str(res))**

**Output**

The original dictionary is : {'gfg': [5, 6, 7, 8], 'is': [10, 11, 7, 5], 'best': [6, 12, 10, 8], 'for': [1, 2, 5]}

The unique values list is : [1, 2, 5, 6, 7, 8, 10, 11, 12]

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

**ANS:-**

**# 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))**

**Output:**

Sum : 600

1. **Write a Python program to Merging two Dictionaries?**

**ANS:-**

**# Python code to merge dict using update() method**

**def Merge(dict1, dict2):**

**return(dict2.update(dict1))**

**# Driver code**

**dict1 = {'a': 10, 'b': 8}**

**dict2 = {'d': 6, 'c': 4}**

**# This returns None**

**print(Merge(dict1, dict2))**

**# changes made in dict2**

**print(dict2)**

**Output:**

None  
{'c': 4, 'a': 10, 'b': 8, 'd': 6}

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

**ANS:-**

**# Python3 code to demonstrate working of**

**# Convert key-values list to flat dictionary**

**# Using dict() + zip()**

**from itertools import product**

**# initializing dictionary**

**test\_dict = {'month' : [1, 2, 3],**

**'name' : ['Jan', 'Feb', 'March']}**

**# printing original dictionary**

**print("The original dictionary is : " + str(test\_dict))**

**# Convert key-values list to flat dictionary**

**# Using dict() + zip()**

**res = dict(zip(test\_dict['month'], test\_dict['name']))**

**# printing result**

**print("Flattened dictionary : " + str(res))**

**Output :**

*The original dictionary is : {‘name’: [‘Jan’, ‘Feb’, ‘March’], ‘month’: [1, 2, 3]} Flattened dictionary : {1: ‘Jan’, 2: ‘Feb’, 3: ‘March’}*

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

**ANS:-**

**# Python code to demonstrate**

**# insertion of items in beginning of ordered dict**

**from collections import OrderedDict**

**# Initialising ordered\_dict**

**iniordered\_dict = OrderedDict([('akshat', '1'), ('nikhil', '2')])**

**# Inserting items in starting of dict**

**iniordered\_dict.update({'manjeet': '3'})**

**iniordered\_dict.move\_to\_end('manjeet', last=False)**

**# Printing result**

**print("Resultant Dictionary : "+str(iniordered\_dict))**

**Output**

Resultant Dictionary : OrderedDict([('manjeet', '3'), ('akshat', '1'), ('nikhil', '2')])

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

**ANS:-**

**# Function to check if string follows order of**

**# characters defined by a pattern**

**from collections import OrderedDict**

**def checkOrder(input, pattern):**

**# create empty OrderedDict**

**# output will be like {'a': None,'b': None, 'c': None}**

**dict = OrderedDict.fromkeys(input)**

**# traverse generated OrderedDict parallel with**

**# pattern string to check if order of characters**

**# are same or not**

**ptrlen = 0**

**for key,value in dict.items():**

**if (key == pattern[ptrlen]):**

**ptrlen = ptrlen + 1**

**# check if we have traverse complete**

**# pattern string**

**if (ptrlen == (len(pattern))):**

**return 'true'**

**# if we come out from for loop that means**

**# order was mismatched**

**return 'false'**

**# Driver program**

**if \_\_name\_\_ == "\_\_main\_\_":**

**input = 'engineers rock'**

**pattern = 'er'**

**print (checkOrder(input,pattern))**

Output:

true

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

**ANS:-**

**myDict = {'ravi': 10, 'rajnish': 9,**

**'sanjeev': 15, 'yash': 2, 'suraj': 32}**

**myKeys = list(myDict.keys())**

**myKeys.sort()**

**sorted\_dict = {i: myDict[i] for i in myKeys}**

**print(sorted\_dict)**

**Output**

{'rajnish': 9, 'ravi': 10, 'sanjeev': 15, 'suraj': 32, 'yash': 2}