▾ ASSIGNMENT / TASK 3

Question on Dictionary-

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

```
markdict = {"Tom":67, "Tina": 54, "Akbar": 87, "Kane": 43, "Divya":73}
print(markdict,"<-original dictionary")
sortdict = sorted(markdict.items(), key=lambda x:x[1])
print(sortdict,"<-Sorted dictionary in ascending order")
sortdict = sorted(markdict.items(), key=lambda x:x[1], reverse=True)
print(sortdict,"<-Sorted dictionary in reverse order")
```

```
    {'Tom': 67, 'Tina': 54, 'Akbar': 87, 'Kane': 43, 'Divya': 73} <-original dictionary
    [('Kane', 43), ('Tina', 54), ('Tom', 67), ('Divya', 73), ('Akbar', 87)] <-Sorted diction
    [('Akbar', 87), ('Divya', 73), ('Tom', 67), ('Tina', 54), ('Kane', 43)] <-Sorted diction
◄                                                                                              ►
```

Q.2 Write a Python Program to add a key to a dictionary.

Sample Dictionary : {0: 10, 1: 20} Expected Result : {0: 10, 1: 20, 2: 30}

```
d={0:"the",1:"day",2:"is"}
print(d," <-This is original dictionary")
d[4]="beautiful"
print(d,"<-This is dictionary after adding a key")
```

```
    {0: 'the', 1: 'day', 2: 'is'}  <-This is original dictionary
    {0: 'the', 1: 'day', 2: 'is', 4: 'beautiful'} <-This is dictionary after adding a key
```

Q.3 Write a program asks for City name and Temperature and builds a dictionary using that Later on you can input City name and it will tell you the Temperature of that City.

```
city_name=input("Enter the city name")
temp=int(input("Enter the temperature of the city"))
d={"city":city_name,"temperature":temp}
print(d)
```

```
    Enter the city nameagra
    Enter the temperature of the city20
    {'city': 'agra', 'temperature': 20}
```

Q. 4 Write a Python program to convert list to list of dictionaries.

Sample lists: ["Black", "Red", "Maroon", "Yellow"], ["#000000", "#FF0000", "#800000", "#FFFF00"]

Expected Output: [{'color_name': 'Black', 'color_code': '#000000'}, {'color_name': 'Red', 'color_code': '#FF0000'}, {'color_name': 'Maroon', 'color_code': '#800000'}, {'color_name': 'Yellow', 'color_code': '#FFFF00'}]

```
color_name=["Black","Maroon","Yellow"]
color_code=["#000","#FF0","#FFF"]
print([{"color_name":i,"color_code":j} for (i,j) in zip(color_name,color_code)])
```

    [{'color_name': 'Black', 'color_code': '#000'}, {'color_name': 'Maroon', 'color_code':

Q. 5 We have following information on Employees and their Salary (Salary is in lakhs),

Employee Salary

John 14

Smith 13

Alice 32

Daneil 21

Using above create a dictionary of Employees and their Salary

Write a program that asks user for three type of inputs,

a.print: if user enter print then it should print all Employees with their Salary in this format, John ==>14 , Smith ==>13 , Alice ==>32 , Daneil ==>21

b.add: if user input adds then it should further ask for an Employee name to add. If Employee already exists in our dataset then it should print that it exists and do nothing. If it doesn't then it asks for Salary and add that new Employee/Salary in our dictionary and print it

c.remove: when user inputs remove it should ask for an Employee to remove. If an Employee exists in our dictionary then remove it and print a new dictionary using format shown above in (a). Else print that Employee doesn't exist!

d. query: on this again ask the user for which Employee he or she wants to query. When a user inputs that Employee it will print the Salary of that Employee.

```
employee=["John","Smith","Alice","Daniel"]
salary=[14,13,32,21]
d=dict(zip(employee,salary))
d
a=input("enter prints/add/query/remove\n")
if a == "add":
```

```
    employee_new=input("Enter the employee name")
    for i in employee:
      if i == employee_new:
        print("Employee already existing")
        break
      else:
        salary_new=int(input("Enter the salary of new employee"))
        d.update({employee_new:salary_new})
        print(d," <-new dictionary after adding")
        break
  if a == "remove":
    employee_new1=input("Enter the emplyee name who is to remove")
    for i in employee:
      if i == employee_new1:
        d.pop(i)
        print(d," <-new dictionary after removine")
        break
      else:
        print("Employee does not exist")
        break
  if a == "query":
    print(d)
    employee_name=input("Which employee to query?")
    for i in employee:
      if i == employee_name:
        print(d.keys(i))
        break
      else:
        print("Employee not found")
        break

      enter prints/add/query/remove
      add
      Enter the employee nameTanisha
      Enter the salary of new employee300
      {'John': 14, 'Smith': 13, 'Alice': 32, 'Daniel': 21, 'Tanisha': 300}  <-new dictionary a
```

Questions on Sets-

Q.1 What is the difference between a set and a frozenset? Create any set and try to use frozenset(setname).

```
sets = {1,2,3,4,5,4} # Duplicate value but not consider
print("Original Set: ",sets)
sets.add(6) # Add new element in set
print("After insert: ",sets)
# set to forzenset
setName = frozenset(sets)
nrint(cotNamo)
```

```
print(setName)
# forzenset to set
setName = set(setName) # use forzenset
print(sets)
```

```
    Original Set:  {1, 2, 3, 4, 5}
    After insert:  {1, 2, 3, 4, 5, 6}
    frozenset({1, 2, 3, 4, 5, 6})
    {1, 2, 3, 4, 5, 6}
```

Q.2 Find the elements in a given set that are not in another set

```
set1 = {10,20,30,40,50}
```

```
set2 = {40,50,60,70,80}
```

Difference between set1 and set2 is {10,20,30}

```
s1=set([10,20,30,40,50])
s2=set([40,50,60,70,80])
print(s1-s2)
```

```
    {10, 20, 30}
```

✓   0s   completed at 17:21   ● ✕