

##Task 3 #GO_STP_342 Task: [https://www.goeduhub.com/11531/dictionary-question-exercise-practice-solutions-learning_\(https://www.goeduhub.com/11531/dictionary-question-exercise-practice-solutions-learning\)](https://www.goeduhub.com/11531/dictionary-question-exercise-practice-solutions-learning_(https://www.goeduhub.com/11531/dictionary-question-exercise-practice-solutions-learning))

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

```
In [1]: import operator
dict1={1:'a',3:'d',2:'f',4:'e',5:'c'}
sorted_dict=dict(sorted(dict1.items(),key=operator.itemgetter(1)))
print("Original Dictionary:",dict1)
print("Ascending order sorted: ",sorted_dict)
rev_sorted_dict=dict(sorted(dict1.items(),key=operator.itemgetter(1),reverse=True))
print("Descending order sorted: ",rev_sorted_dict)
```

```
Original Dictionary: {1: 'a', 3: 'd', 2: 'f', 4: 'e', 5: 'c'}
Ascending order sorted: {1: 'a', 5: 'c', 3: 'd', 4: 'e', 2: 'f'}
Descending order sorted: {2: 'f', 4: 'e', 3: 'd', 5: 'c', 1: 'a'}
```

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

```
In [2]: dict1.update({12:'30'})
print(dict1)
```

```
{1: 'a', 3: 'd', 2: 'f', 4: 'e', 5: 'c', 12: '30'}
```

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.

```
In [3]: city_temp={}
print (" Enter q to exit data entry")
while True:
    city=input("Enter city name: ").upper()
    if city == 'Q':
        break
    else:
        temp=eval(input("Enter temperature: "))
        if temp == 'Q':
            break
        city_temp[city]=temp

display_city= input(' Enter the name of the city to find Temperature ').upper()

if (len(city_temp)==0):
    print('The dictionary is empty')
elif display_city not in city_temp.keys():
    print('Entered City is not present in dictionary')
else:
    print('Temperature of ', display_city, 'is ',city_temp[display_city])
```

```
Enter q to exit data entry
Enter city name: Delhi
Enter temperature: 39
Enter city name: kolkata
Enter temperature: 35
Enter city name: mumbai
Enter temperature: 36
Enter city name: chennai
Enter temperature: 39
Enter city name: pune
Enter temperature: 31
Enter city name: q
Enter the name of the city to find Temperature pune
Temperature of PUNE is 31
```

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'}]

```
In [4]: clr_nm_lst=["Black", "Red", "Maroon", "Yellow"]
clr_cd_lst= ["#000000", "#FF0000", "#800000", "#FFFF00"]
color=[]
for i in range(0, len(clr_nm_lst)):
    color.append({'color_name':clr_nm_lst[i], 'color_code':clr_cd_lst[i]})
color
```

```
Out[4]: [{'color_name': 'Black', 'color_code': '#000000'},
{'color_name': 'Red', 'color_code': '#FF0000'},
{'color_name': 'Maroon', 'color_code': '#800000'},
{'color_name': 'Yellow', 'color_code': '#FFFF00'}]
```

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

1. Using above create a dictionary of Employees and their Salary Write a program that asks user for three type of inputs, 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
2. 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
3. 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!
4. 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.

```

In [6]: emp_dict = {'John' : 14, 'Smith' : 13, 'Alice' : 32, 'Daneil' : 21}

def print_rec():
    print('Output Records:')
    print('\tEmployee\tSalary')
    for rec in emp_dict.items():
        print('\t',rec[0], ' ==> ',rec[1])

def add_rec():
    emp_name = input('Enter Employee Name: ').capitalize()
    if emp_name in emp_dict.keys():
        print('The Employee already exists!')
    else:
        emp_salary = eval(input('The Salary of the Employee : '))
        emp_dict.update({emp_name : emp_salary})

def remove_rec():
    emp_name = input('Enter Employee Name for removal: ').capitalize()
    if emp_name in emp_dict.keys():
        emp_dict.pop(emp_name)
        print('The Employee record is removed!')
    else:
        print('The Employee record does not exist!')

def query_rec():
    emp_name = input('Enter the Employee Name : ').capitalize()
    if emp_name in emp_dict.keys():
        print('The Salary of Given Employee : ',emp_dict[emp_name])
    else:
        print('The Given Employee Not Exists !')

while True:
    print('')
    print('1) Print All the records : ')
    print('2) Add New Record : ')
    print('3) Remove exists Record : ')
    print('4) Query Single Record: ')
    print('5) Exit')
    option = eval(input('Enter the Option : '))
    if option == 1:
        print_rec()
    elif option == 2:
        add_rec()
        print_rec()
    elif option == 3:
        remove_rec()
        print_rec()
    elif option == 4:
        query_rec()
    elif option == 5:
        break
    else:
        print('Invalid Option')

```

```

1) Print All the records :
2) Add New Record :
3) Remove exists Record :
4) Query Single Record:
5) Exit
Enter the Option : 1
Output Records:

```

Employee	Salary
John ==>	14
Smith ==>	13
Alice ==>	32
Daneil ==>	21

1) Print All the records :
2) Add New Record :
3) Remove exists Record :
4) Query Single Record:
5) Exit
Enter the Option : 2
Enter Employee Name: sai
The Salary of the Employee : 13
Output Records:

Employee	Salary
John ==>	14
Smith ==>	13
Alice ==>	32
Daneil ==>	21
Sai ==>	13

1) Print All the records :
2) Add New Record :
3) Remove exists Record :
4) Query Single Record:
5) Exit
Enter the Option : 3
Enter Employee Name for removal: sai
The Employee record is removed!
Output Records:

Employee	Salary
John ==>	14
Smith ==>	13
Alice ==>	32
Daneil ==>	21

1) Print All the records :
2) Add New Record :
3) Remove exists Record :
4) Query Single Record:
5) Exit
Enter the Option : 4
Enter the Employee Name : alice
The Salary of Given Employee : 32

1) Print All the records :
2) Add New Record :
3) Remove exists Record :
4) Query Single Record:
5) Exit
Enter the Option : 6
Invalid Option

1) Print All the records :
2) Add New Record :
3) Remove exists Record :
4) Query Single Record:
5) Exit
Enter the Option : 5

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

A: Frozen set is just an immutable version of a Python set object. While elements of a set can be modified at any time, elements of the frozen set remain the same after creation.

```
In [7]: week = ('Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat')
```

```
fSet = frozenset(week)
print('The frozen set is:', fSet)
print('The empty frozen set is:', frozenset())

fSet.add('Nov')
```

```
The frozen set is: frozenset({'Fri', 'Tue', 'Sat', 'Wed', 'Mon', 'Sun', 'Thu'})
The empty frozen set is: frozenset()
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-7-77aabfa6a41f> in <module>
      5 print('The empty frozen set is:', frozenset())
      6
----> 7 fSet.add('Nov')
```

AttributeError: 'frozenset' object has no attribute 'add'

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}
```

```
In [8]: set1 = {10, 20, 30, 40, 50}
set2 = {40, 50, 60, 70, 80}
set1.difference_update(set2)
print(' Difference between set1 and set2 is', set1 )
```

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

```
In [ ]:
```