a) Display name and phone number of all your friends
b) Search for a friends name in your list , if found display their phone number
c) Delete a particular friend
d) Modify the phone number of an existing friend
e) Display the dictionaries in sorted order of friends

2. Write a program to create a dictionary from a String which tracks down no of items a character is present in the string

3.Write a program to read email IDs of n number of students and store them in a tuple. Create two new tuples, one to store only the usernames from the email ids and second to store the domain names from the email ids. Print all three tuples at the end of the program.[Hint : you may use the function split()]

4. Write a program to print the frequency of a number accepted from the user in the given tuple.

5. Create a dictionary which contains dice combinations as(tuple) keys and the prizes awarded as values.
Write a program to take lot or ask user to enter the dice combination, check in the dictionary and print appropriate message

DICTIONARIES

CONTENT – REVIEW

Dictionary

- Unordered collection of items / elements. Each item has two parts - key : value
- A mapping between key and it's value, separated by colon (:)

- Items are enclosed in {} and separated by comma.
- Are optimized to retrieve data using key. So key (values) should be unique.
- They can be integer, float, string or tuple.
- D[key] can be used for accessing element, also to add element in an existing dictionary

Creation- Using

- built - in function dict()
- assignment operator creates deep copy
- copy() creates shallow copy
- {} to create blank dictionary

Shallow and Deep Copy

```
d4=d3.copy()   #shallow copy  changes done in d3 does not reflect in d4
print(d4)
```

```
{1: 'Anamika', 2: 'Sooraj'}
```

```
d3[3]="Shyam"
print(d3)
print(d4)
```

```
{1: 'Anamika', 2: 'Sooraj', 3: 'Shyam'}
{1: 'Anamika', 2: 'Sooraj'}
```

```
d5=d3  #deep copy...changes done later in d3 reflects d5
d3[4]="Viji"
print(d3)
print(d5)
```

```
{1: 'Anamika', 2: 'Sooraj', 3: 'Shyam', 4: 'Viji'}
{1: 'Anamika', 2: 'Sooraj', 3: 'Shyam', 4: 'Viji'}
```

Accessing Items

```
#mapping data type...unordered collection of items
#item key:value
#key-   immutable(int float complex bool string tuple ) , unique
#value- anything..nested, repeated

prize={"first":"books","second":"pencil box","third":100,True:"Participated",False:"Din take part"}
#
print(type(prize.keys()),": ",prize.keys())
mykeys=list(prize.keys())
print(type(mykeys),":",mykeys)

print(type(prize.values()),": ",prize.values())
myval=list(prize.values())
print(type(myval),":",myval)
```

```
<class 'dict_keys'> :  dict_keys(['first', 'second', 'third', True, False])
<class 'list'> : ['first', 'second', 'third', True, False]
<class 'dict_values'> :  dict_values(['books', 'pencil box', 100, 'Participated', 'Din take part'])
<class 'list'> : ['books', 'pencil box', 100, 'Participated', 'Din take part']
```

Accessing Items…

```
print(type(prize.items()),": ",prize.items())
myitem=list(prize.items())
print(type(myitem),":",myitem)   #list of tuples(key,value)
```

```
<class 'dict_items'> :  dict_items([('first', 'books'), ('second', 'pencil box'), ('third', 100), (Tr
ue, 'Participated'), (False, 'Din take part')])
<class 'list'> : [('first', 'books'), ('second', 'pencil box'), ('third', 100), (True, 'Participate
d'), (False, 'Din take part')]
```

```
#access the value through key
print(prize["first"])  #use []
```

```
books
```

```
print(prize.get("first"))  #get()
```

```
books
```

Adding an item

```
#add an item in a dict
prize["last"]="pen"   #dict[key]=value
prize
```

```
{'first': 'books',
 'second': 'pencil box',
 'third': 100,
 True: 'Participated',
 False: 'Din take part',
 'last': 'pen'}
```

Updating an Item

```
d1={"fourth":500}
prize.update(d1)   #use existing dict in update
prize.update({"fifth":300})   #take an instance of dict in update
prize
```

```
{'first': 'books',
 'second': 'pencil box',
 'third': 100,
 True: 'Participated',
 False: 'Din take part',
 'last': 'pen',
 'fourth': 500,
 'fifth': 300}
```

```
#modify the value   new keys-new item will added, existing key..vale wil be updated
prize["first"]="DVD"   #change the value of first as DVD   dict[key]=value...key should be
prize.update({"second":"geometry box"})
prize
```

```
{'first': 'DVD',
 'second': 'geometry box',
 'third': 100,
 True: 'Participated',
 False: 'Din take part',
 'last': 'pen',
 'fourth': 500,
 'fifth': 300}
```

Traversing

```
#traversing  - for loop with key
print("POSITION\t:\tPRIZE")
for key in prize:
    print(key,"\t\t\t",prize[key])
```

```
POSITION        :       PRIZE
first                    DVD
second                   geometry box
third                    100
True                     Participated
False                    Din take part
last                     pen
fourth                   500
fifth                    300
```

```
#traversing  - for loop with key,value pairs
print("POSITION\t:\tPRIZE")
for key,value in prize.items():
    print(key,"\t\t\t",value)
```

```
POSITION        :       PRIZE
first                    DVD
second                   geometry box
third                    100
True                     Participated
False                    Din take part
last                     pen
fourth                   500
fifth                    300
```

Built – in Functions

Function/Method	Purpose	Example
len()	This function returns the number of elements i.e. the key-value pairs present in the dictionary.	>>>d={1: 'Amrit', 2: 'Bhavesh', 3: 'Chetan', 4: 'Falguni', } >>> len(d) 6
d.items()	This function returns the list of elements in the dictionary.	>>> d.items() dict_items([(1, 'Amrit'), (2, 'Bhavesh'), (3, 'Chetan'), (4, 'Falguni')])
d.keys()	This function returns the all the keys that are present in the dictionary.	>>> d.keys() dict_keys([1, 2, 3, 4])
d.values()	This returns all the values in the dictionary.	>>> d.values() dict_values(['Amrit', 'Bhavesh', 'Chetan', 'Falguni'])

Function/Method	Purpose	Example

d.get(key)	This function returns the value corresponding to a key in the dictionary. If the key is not present, the function returns, 'None'.	>>> d.get(2)
		'Bhavesh'
		
		>>> d.get(8, -1)
		-1

d.update(object)	This function merges the object enclosed in brackets with d.	>>> d2={5: 'Kartikay', 6: 'Prerna'}
		>>> d.update(d2)
		>>> d
		{1: 'Amrit', 2: 'Bhavesh', 3: 'Chetan', 4: 'Falguni', 5: 'Kartikay', 6: 'Prerna'}

d.clear()	This function clears the entire dictionary. It deletes all the key-value pairs.	>>>d2={1:"Deer", 2:"Bear", 3:"Cat", 4:"Elephant"} >>>d2 {1:"Deer", 2:"Bear", 3:"Cat", 4:"Elephant"} >>>d2.clear() >>>d2 {}

Built – in Functions

Function/Method	Purpose	Example

d.pop(key)	This function removes a key along with its value in a dictionary.	>>> d2={1:"Apple", 2:"Ball", 3:"Pineapple",4:"Mangoes"}
		
		>>> d2.pop(2)
		'Ball'
		>>> d2
		{1: 'Apple', 3: 'Pineapple', 4: 'Mangoes'}

A Sample code to find frequency of vowel letters

```
#dictionary based...
phrase=list("Everything will be fine,All the best")
phrase
#dict ..track of vowel letters count
#vowel={'a':4,'i':5}
vowel={}
a=phrase.count('a')
a=a+phrase.count('A')
vowel['a']=a
i=phrase.count('i')
i=i+phrase.count('I')
vowel['i']=i
e=phrase.count('e')
e=e+phrase.count('E')
vowel['e']=e
o=phrase.count('o')
o=o+phrase.count('O')
vowel['o']=o
u=phrase.count('u')
u=u+phrase.count('U')
vowel['u']=u
print(vowel)
```

```
{'a': 1, 'i': 3, 'e': 6, 'o': 0, 'u': 0}
```

A Sample code to draw lot

```
#lottery system
import random
lot={(3,2,2):"First",(1,4,3):"Second",(6,1,6):"Third",(3,1,5):"Fourth"}
prize={"First":"books","Second":"pencil box","Third":100,"Fourth":"chocolate"}
for i in range(10):
    a=random.randint(1,6)
    b=random.randint(1,6)
    c=random.randint(1,6)
    dice=a,b,c
    if dice in lot:
        n=lot[dice]#1st-4th
        print("Congrats, you won ",prize[n])
```

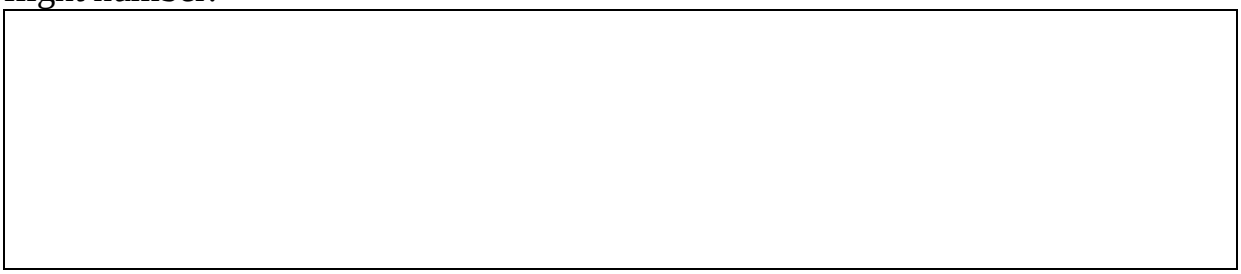
```
Congrats, you won  books
Congrats, you won  pencil box
```

WORKSHEETS

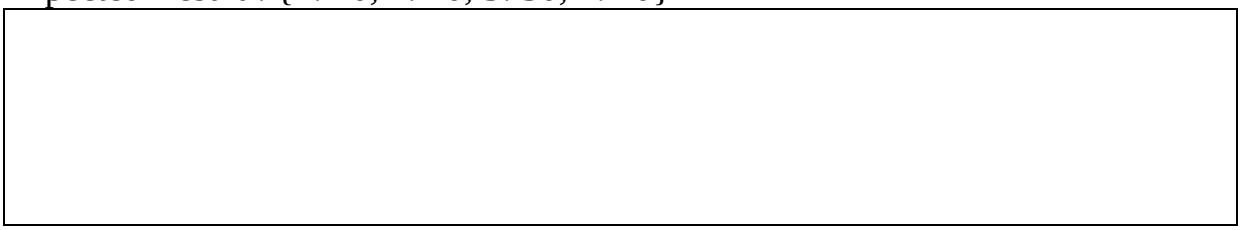
I. Write Python statements to:

1. Create a dictionary which contains the name of airlines corresponding to the

flight number.



b. Concatenate following dictionaries to create a new one
dic1={1:10, 2:20}
dic2={3:30, 4:40}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40}



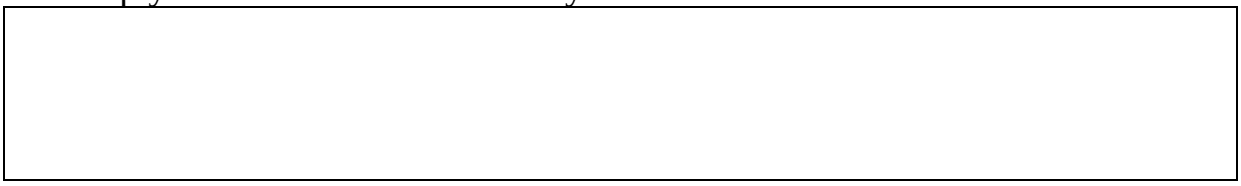
3 Add a key to a dictionary.

4 sum all the items in a dictionary

5 Sort (ascending and descending) a dictionary by value.

6 check if a given key already exists in a dictionary

7. multiply all the items in a dictionary



8. get the maximum and minimum value in a dictionary

9 print all unique values in a dictionary

10. check a dictionary is empty or not

11. print a dictionary in table format

II . What will be the output of the following code?

1.

```
dictionary = {"Roll":9 , "Class":8, "Section":'A', "Marks":90}
print("Roll" in dictionary)
```

2.

```
b. a = {}
a[1] = 6
a['1'] = 2
a[1]= a[1]+1
count = 0
for i in a:
   count += a[i]
print(count)
```

CYBER SAFETY / SECURITY /
SOCIETY LAW AND ETHICS

CONTENT – REVIEW

Society, Law and Ethics