**EXERCISE NO** :9

**DATE** : 22-11-2020

**AIM:**

To write and run a Python program to fill in the desired output.

**PROGRAM:**

print('\n—dictionaries')  **#Output:** -- dictionaries

d = {'a': 1, 'b': 2}

print(d['a']) **#Output:** 1

del d['a']

# iterate  
d = {'a': 1, 'b': 2}  
for key, value in d.items():  
 print(key, ':', value)

for key in d:  
 print(key, d[key])

# d.fromkeys(iterable[,value=None]) ->dict: with keys from iterable and all same value  
d = d.fromkeys(['a', 'b'], 1)  
print(d) **#Output:** {'a': 1, 'b': 1}

# d.clear() -> removes all items from d  
d = {'a': 1, 'b': 2}  
d.clear()  
print(d) **#Output:** {}

# d.items() -> list: copy of d's list of (key, item) pairs  
d = {'a': 1, 'b': 2}  
print(d.items()) **#Output:** [('a', 1), ('b', 2)]

# d.keys() -> list: copy of d's list of keys   
d = {'a': 1, 'b': 2}  
print(d.keys()) **#Output:** ['a', 'b']

# d.values() -> list: copy of d's list of values  
 d = {'a': 1, 'b': 2}  
 print(d.values())  **#Output:** [1, 2]

# d.get(key,defval) -> value: d[key] if key in d, else defval  
d = {'a': 1, 'b': 2}  
print(d.get("c", 3))  **#output:** 3

print(d)  **#output:** {'a'

# d.setdefault(key[,defval=None]) -> value: if key not in d set d[key]=defval, return d[key]  
d = {'a': 1, 'b': 2}  
print('d.setdefault("c", []) returns ' + str(d.setdefault("c", 3)) + ' d is now ' + str(d))

**#Output:** d.setdefault("c", []) returns 3 d is now {'a'

#d.pop(key[,defval]) -> value: del key and returns the corresponding value. If key is not found, defval is returned if given, otherwise KeyError is raised  
d = {'a': 1, 'b': 2}  
print('d.pop("b", 3) returns ' + str(d.pop("b", 3)) + ' d is now ' + str(d))

**#Output:** d.pop("b", 3) returns 2 d is now {'a'

print('d.pop("c", 3) returns ' + str(d.pop("c", 3)) + ' d is still ' + str(d))  
**#Output:** d.pop("c", 3) returns 3 d is still {'a'

# sort on values  
import operator  
x = {1: 4, 5: 4, 4: 4}  
sorted\_x = sorted(x.items(), key=operator.itemgetter(1), reverse=True)

**#output:** print('sorted(x.items(), key=operator.itemgetter(1)) sorts on values ' + str(sorted\_x))

# max of values  
d = {'a':1000, 'b':3000, 'c': 100}  
print('key of max value is ' + max(d.keys(), key=(lambda key: d[key])))

**#Output:** key of max value is b

**RESULT:**

Hence, the program is executed successfully