PYTHON (2nd)

BY- Harsh Rai

ROLLNO – 2215000750
CLASS – B-TECH (CS)
SECTION – (U-20)

Let’s Begin With Dictionaries In Python.

Q-1. What Will Be The Output Of The Following Code Snippet?

```
a = {(1,2):1,(2,3):2}

print(a[1,2])
```

A. Key Error
B. 1
C. {(2,3):2}
D. {(1,2):1}

ANS=B. 1

Q-2. What Will Be The Output Of The Following Code Snippet?

```
a = {'a':1,'b':2,'c':3}

 print (a['a','b'])

 print(a.get('a','b'))
```

A. Key Error
B. [1,2]
C. {'a':1,'b':2}
D. (1,2)

ANS= print (a['a','b']) KeyError: ('a', 'b')

Q-3. What Will Be The Output Of The Following Code Snippet?

```
fruit = {}


def addone(index):

    if index in fruit:
        fruit[in

    dex]   +=

    1 else:

        fruit[index] = 1


addone('A

pple')
```

```
addone('B

anana')

addone('a

pple')

print (len(fruit))
```

A. 1
B. 2
C. 3
D. 4

ANS= C 3

Q-4. What Will Be The Output Of The Following Code Snippet?

```
arr = {}

arr[1] = 1

arr['
1']
= 2

arr[1] += 1



sum = 0

for k in arr:

    sum += arr[k]



print (sum)
```

A. 1
B. 2
C. 3
D. 4

ANS= D. 4

Q-5. What Will Be The Output Of The Following Code Snippet?

```
my_dict = {}

my_dict[1] = 1

my_dict['1'] = 2

my_dict[1.0] = 4


sum = 0

for k in my_dict:

    sum += my_dict[k]


print (sum)
```

A. 7
B. S
y
n
t
a
x
e
r
r
o
r
C. 3
D
.
6

ANS= D.6

Q-6. What Will Be The Output Of The Following Code Snippet?

```
my_dict = {}

my_dict[(1,2,4)] = 8

my_dict[(4,2,1)] = 10
my_dict[(1,2)] = 12



sum = 0

for k in my_dict:

    sum += my_dict[k]



print (sum)

print(my_

dict)
```

A. Syntax error
B. 30
{(1, 2): 12, (4, 2, 1): 10, (1, 2, 4): 8}
C. 47
{(1, 2): 12, (4, 2, 1): 10, (1, 2, 4): 8}
D. 30
{[1, 2]: 12, [4, 2, 1]: 10, [1, 2, 4]: 8}

.ANS= A. SYNTAX ERROR
30
{(1, 2, 4): 8, (4, 2, 1): 10, (1, 2): 12}

Q-7. What Will Be The Output Of The Following Code Snippet?

b

o

x

=

{

}

j

a

r

s

=

{

}

c

r

a

t

e

s

=

{

}

box['biscuit'] = 1

box['cake'] = 3

jars

['ja

m']

= 4

cra

tes

['bo

x']

=

bo

x

cra

tes

['jar

s']

=

jar

s

pri

nt

(le

n(c

rat

es[

bo

x]))

A. 1
B. 3
C. 4
D. T
y
p
e
E
r
r
o
r

ANS= D. TypeError: unhashable type: 'dict'

Q-8. What Will Be The Output Of The Following Code Snippet?

dict = {'c': 97, 'a': 96, 'b': 98}

for _ in

sorted(

dict):

print

(dict[_])

A. 96 98 97
B. 96 97 98
C. 98 97 96
D. NameError

ANS= **A.** 96 98 97

Q-9. What Will Be The Output Of The Following

Code Snippet? rec = {"Name" : "Python",

"Age":"20"} r = rec.copy() print(id(r) == id(rec))

A. True
B. False
C. 0 **D.** 1

ANS= B. False

Q-10. What Will Be The Output Of The Following Code Snippet?

rec = {"Name" : "Python", "Age":"20", "Addr" : "NJ",

"Country" : "USA"} id1 = id(rec)

del rec

rec = {"Name" : "Python", "Age":"20", "Addr" : "NJ",

"Country" : "USA"} id2 = id(rec)

print(id1 == id2)

A
.
T
r
u
e
B
.
F
a
l
s
e
C. 1
D. Exception

ANS = A. TRUE

Python Dictionary [38 exercises]

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

ANS =CODE
```
import operator d = {1: 2,  3:  4,  4:  3,  2:  1,  0:  0}
print('Original dictionary : ',d)
sorted_d = sorted(d.items(), key=operator.itemgetter(1))
print('Dictionary in ascending order by value : ',sorted_d)
sorted_d = dict( sorted(d.items(),
key=operator.itemgetter(1),reverse=True))
print('Dictionary in descending order by value : ',sorted_d)
```

OUTPUT=

```
Original dictionary :  {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
Dictionary in ascending order by value :  [(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)]
Dictionary in descending order by value :  {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}
```

2. Write a Python script to add a key to a dictionary

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

ANS=
```
d = {0:10, 1:20}
print(d)
d.update({2:30})
print(d)
```

OUTPUT=

```
{0: 10, 1: 20}
{0: 10, 1: 20, 2: 30}
```

3. Write a Python script to concatenate following dictionaries to create a new one.

Sample Dictionary : dic1={1: 10, 2:20 } dic2={3: 30, 4:40 } dic3={5: 50,6: 60}

Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

ANS= CODE

```
dic1={1:10, 2:20} dic2={3:30,
4:40} dic3={5:50,6:60} dic4 =
{} for d in (dic1, dic2, dic3):
dic4.update(d) print(dic4)
```

OUTPUT=

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

4. Write a Python script to check if a given key already exists in a dictionary.

ANS= CODE

```
d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6:

60} def is_key_present(x):   if x in d:

      print('Key is present in the dictionary')

else:

      print('Key is not present in the dictionary')
is_key_present(5) is_key_present(9)
```

OUTPUT=

```
Key is present in the dictionary
Key is not present in the dictionary
```

5. Write a Python program to iterate over dictionaries using for loops.

ANS = CODE

```
d = {'Red': 1, 'Green': 2, 'Blue': 3}
for color_key, value in d.items():
    print color_key, 'corresponds to ', d[color_key]
```

OUTPUT=

```
Red corresponds to  1
Green corresponds to  2
Blue corresponds to  3
```

6. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x

Sample Dictionary (n = 5) :
Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

ANS = CODE

```
n=int(input("Input a number ")) d

= dict()

 for x in

range(1,n+1):

    d[x]=x*x
print(d)
```

7. Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys.

Sample Dictionary

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169,
14: 196, 15: 225}

ANS

=CODE

```
for   x i
    d [x     ]=x
 print (d     )
d=dict()
range(1,1
6):
          **2
```

8. Write a Python script to merge two Python dictionaries.

ANS= CODE

```
   d1 = {'a':
100, 'b': 200}
d2 = {'x': 300,
'y': 200} d =
d1.copy()
d.update(d2)
print(d) OUTPUT
=
{'x': 300, 'y': 200, 'a': 100, 'b': 200}
```

9. Write a Python program to iterate over dictionaries using for loops.
ANS= CODE

```
d = {'Red': 1, 'Green':
2, 'Blue': 3}  for      d.items
color_key, value in ():
     print(color_key, 'corresponds to ',
d[color_key]) OUTPUT
Red corresponds to  1
Green corresponds to  2
Blue corresponds to  3
```

10. Write a Python program to sum all the items in a dictionary.
ANS = CODE

```
my_dict = {'data1':100,'data2':-54,'data3':247}

print(sum(my_dict.values()))
```

Output:293

11. Write a Python program to multiply all the items in a dictionary.
ANS= CODE

```
my_dict = {'data1':100,'data2':-

54,'data3':247} result=1 for key in

my_dict:


    result=result * my_dict[key]


print(result)
```

Output:-1333800

12. Write a Python program to remove a key from a dictionary.

ANS= CODE

```
myDict = {'a':1,'b':2,'c':3,'d':4}

print(myDict)

if 'a' in myDict:

    del myDict['a']

print(myDict)
```

Output:

```
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
{'b': 2, 'c': 3, 'd': 4}
```

13. Write a Python program to map two lists into a dictionary. ANS =

```
keys = ['red', 'green', 'blue']

values = ['#FF0000','#008000', '#0000FF']

color_dictionary = dict(zip(keys, values))

print(color_dictionary)
```

Output: `{'red': '#FF0000', 'green': '#008000', 'blue': '#0000FF'}`

14. Write a Python program to sort a dictionary by key.
ANS=

15. Write a Python program to get the maximum and minimum value in a dictionary.

```
color_dict = {'red':'#FF0000',
          'green':'#008000',
          'black':'#000000',
          'white':'#FFFFFF'}


for key in sorted(color_dict):
    print("%s: %s" % (key, color_dict[key]))
```

Output:

```
black: #000000
green: #008000
red: #FF0000
white: #FFFFFF
```

ANS=

```
my_dict = {'x':500, 'y':5874, 'z': 560}


key_max = max(my_dict.keys(), key=(lambda k: my_dict[k]))
key_min = min(my_dict.keys(), key=(lambda k: my_dict[k]))


print('Maximum Value: ',my_dict[key_max])
print('Minimum Value: ',my_dict[key_min])
```

Output:

```
Maximum Value:  5874
Minimum Value:  500
```

16. Write a Python program to get a dictionary from an object's fields.

ANS=

```
class

dictObj(object

):      def

__init__(self)

:

self.x = 'red'
```

```
        self.y = 'Yellow'

self.z = 'Green'      def

do_nothing(self):

        pass test =
dictObj()
print(test.__dict__)
```

Output:

```
{'x': 'red', 'y': 'Yellow', 'z': 'Green'}
```

17. Write a Python program to remove duplicates from Dictionary. ANS=

```
test_dict = { 'gfg' : 10, 'is' : 15, 'best' : 20,
'for' :
10, 'geeks' : 20} print("The original dictionary

is : " + str(test_dict)) temp = [] res = dict() for

key, val in test_dict.items():     if val not in

temp:         temp.append(val)         res[key]

= val
```

```
print("The dictionary after values removal : "
+ str(res))
```

OUTPUT=
The original dictionary is : {‘gfg’: 10, ‘for’: 10, ‘geeks’: 20, ‘is’: 15, ‘best’: 20}
The dictionary after values removal : {‘gfg’: 10, ‘geeks’: 20, ‘is’: 15}

18. Write a Python program to check a dictionary is empty or not.
ANS=

```
my_dict = {}

if not

bool(my_dict):

    print("Dictionary is empty")
```

Output: `Dictionary is empty`

19. Write a Python program to combine two dictionary adding values for common keys.
d1 = {'a': 100, 'b': 200, 'c':300} d2 = {'a': 300, 'b': 200, 'd':400}
Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})
ANS=

```
from collections import Counter

d1 = {'a': 100, 'b': 200, 'c':300}

d2 = {'a': 300, 'b': 200, 'd':400}

d = Counter(d1) + Counter(d2)

print(d)
```

Output: `Counter({'b': 400, 'd': 400, 'a': 400, 'c': 300})`

20. Write a Python program to print all unique values in a dictionary. Sample Data : [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S005"}, {"V":"S009"},{"VIII":"S007"}]
Expected Output : Unique Values: {'S005', 'S002', 'S007', 'S001', 'S009'}

ANS=

```
L = [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"},
{"VII":"S005"},

{"V":"S009"},{"VIII":"S007"}]

print("Original List: ",L) u_value = set(

val for dic in L for val in dic.values())

print("Unique Values: ",u_value)
```

Output:

```
Original List:  [{'V': 'S001'}, {'V': 'S002'}, {'VI': 'S001'},
{'VI': 'S005'}, {'VII': 'S005'}, {'V': 'S009'},
 {'VIII': 'S007'}]
Unique Values:  {'S009', 'S002', 'S007', 'S005', 'S001'}
```

21. Write a Python program to create and display all combinations of letters, selecting each letter from a different key in a dictionary. Sample data : {'1':['a','b'], '2':['c','d']}

Expected Output:
ac
ad
bc
bd

22. Write a Python program to find the highest 3 values in a dictionary.
ANS=

```
import itertools       d ={'1':['a','b'], '2':['c','d']} for combo

in itertools.product(*[d[k] for k in sorted(d.keys())]):

    print(''.join(combo)) Output:
```

```
ac
ad
bc
bd
```

23. Write a Python program to combine values in python list of dictionaries.
Sample data: [{'item': 'item1', 'amount': 400}, {'item': 'item2', 'amount': 300}, {'item': 'item1', 'amount': 750}]
Expected Output: Counter({'item1': 1150, 'item2': 300})
ANS=

```
from collections import Counter

item_list = [{'item': 'item1', 'amount': 400}, {'item': 'item2',
'amount': 300}, {'item': 'item1', 'amount': 750}]

result = Counter() for d in item_list:

    result[d['item']] += d['amount']

print(result)
```

Output: `Counter({'item1': 1150, 'item2': 300})`

24. Write a Python program to create a dictionary from a string.
Note: Track the count of the letters from the string.
Sample string : 'w3resource'
Expected output: {'3': 1, 's': 1, 'r': 2, 'u': 1, 'w': 1, 'c': 1, 'e': 2, 'o': 1}
ANS=

```
from collections import defaultdict,

Counter str1 = 'w3resource'  my_dict = {}

for letter in str1:

    my_dict[letter] = my_dict.get(letter, 0) + 1

print(my_dict)
```

Output:`{'w': 1, '3': 1, 'r': 2, 'e': 2, 's': 1, 'o': 1, 'u': 1, 'c': 1}`

25. Write a Python program to print a dictionary in table format.
ANS=

```
my_dict = {'C1':[1,2,3],'C2':[5,6,7],'C3':[9,10,11]} for row in zip(*([key] +

(value) for key, value in sorted(my_dict.items()))):

    print(*row)
```

Output:

```
C1 C2 C3
1 5 9
2 6 10
3 7 11
```

26. Write a Python program to count the values associated with key in a dictionary.
Sample data: = [{'id': 1, 'success': True, 'name': 'Lary'}, {'id': 2, 'success': False, 'name':
'Rabi'}, {'id': 3, 'success': True, 'name': 'Alex'}]
Expected result: Count of how many dictionaries have success as True ANS=

```
student = [{'id': 1, 'success': True, 'name': 'Lary'},

 {'id': 2, 'success': False, 'name': 'Rabi'},
{'id': 3, 'success': True, 'name': 'Alex'}]
print(sum(d['id'] for d in student))
```

```
print(sum(d['success'] for d in
student)) Output:
6
2
```

27. Write a Python program to convert a list into a nested dictionary of keys.
ANS=

```
num_list = [1, 2, 3, 4]

new_dict = current = {}

for name in num_list:

current[name] = {}

current = current[name]

print(new_dict)
```

Output: `{1: {2: {3: {4: {}}}}}`

28. Write a Python program to sort a list alphabetically in a dictionary.
ANS=

```
num = {'n1': [2, 3, 1], 'n2': [5, 1, 2], 'n3': [3, 2, 4]}

sorted_dict = {x: sorted(y) for x, y in num.items()}

print(sorted_dict)
```

Output: `{'n1': [1, 2, 3], 'n2': [1, 2, 5], 'n3': [2, 3, 4]}`

29. Write a Python program to remove spaces from dictionary keys.
ANS=

```
student_list = {'S  001': ['Math', 'Science'], 'S    002': ['Math',
'English']} print("Original dictionary:

",student_list)

student_dict = {x.translate({32: None}): y for x, y
in student_list.items()} print("New dictionary:
",student_dict)
```

Output:

```
Original dictionary:  {'S  001': ['Math', 'Science'], 'S    002':
['Math', 'English']}
New dictionary:  {'S001': ['Math', 'Science'], 'S002': ['Math',
'English']}
```

30. Write a Python program to get the top three items in a shop. Sample data: {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24} Expected Output: item4 55 item1 45.5 item3 41.3

```
from heapq import nlargest

from operator import itemgetter

items = {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55,
'item5': 24}

for name, value in nlargest(3, items.items(), key=itemgetter(1)):

    print(name, value)
```

Output:

```
item4 55
item1 45.5
item3 41.3
```

31. Write a Python program to get the key, value and item in a dictionary.

ANS=

```
dict_num = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

print("key   value   count") for count, (key, value) in

enumerate(dict_num.items(), 1):

    print(key,'   ',value,'    ', count)
```

Output:

```
key  value  count
1     10      1
2     20      2
3     30      3
4    40      4
5    50      5
6     60      6
```

32. Write a Python program to print a dictionary line by line.

ANS=

```
students = {'Himanshu':{'class':'btech',
        'rolld_id':2},
        'Shewta':{'class':'btech',
        'roll_id':3}} for
a in students:
    print(a)     for b in
students[a]:
        print (b,':',students[a][b])
```

Output:

```
Himanshu
class : btech
rolld_id : 2
Shewta
class : btech
roll_id : 3
```

33. Write a Python program to check multiple keys exists in a dictionary.
Ans=

```
student = {
  'name': 'Himanshu',
  'class': 'btech',
  'roll_id': '2'
} print(student.keys() >=
{'class', 'name'})
print(student.keys() >= {'name',
'Himanshu'}) print(student.keys()
>= {'roll_id', 'name'})
```

Output:

```
True
False
True
```

34. Write a Python program to count number of items in a dictionary value that is a list. Ans

```
dict =  {'Alex': ['subj1', 'subj2', 'subj3'], 'David': ['subj1',
'subj2']}

ctr = sum(map(len, dict.values()))

print(ctr)
```

Output 5

35. Write a Python program to sort Counter by value.
Sample data : {'Math':81, 'Physics':83, 'Chemistry':87}
Expected data: [('Chemistry', 87), ('Physics', 83), ('Math', 81)] Ans=

```
from collections import Counter x = Counter({'Math':81,

'Physics':83, 'Chemistry':87}) print(x.most_common())
```

Output:`[('Chemistry', 87), ('Physics', 83), ('Math', 81)]`

36. Write a Python program to create a dictionary from two lists without losing duplicate values.
Sample lists: ['Class-V', 'Class-VI', 'Class-VII', 'Class-VIII'], [1, 2, 2, 3]
Expected Output: defaultdict(<class 'set'>, {'Class-VII': {2}, 'Class-VI': {2}, 'Class-VIII': {3}, 'Class-V': {1}})
ANS=

```
from collections import defaultdict

class_list = ['Class-V', 'Class-VI', 'Class-VII', 'Class-VIII']

id_list = [1, 2, 2, 3]

temp = defaultdict(set)

for c, i in zip(class_list, id_list):

    temp[c].add(i)

print(temp)
```

Output:
```
defaultdict(<class 'set'>, {'Class-V': {1}, 'Class-VI': {2}'Class-VII': {2}, 'Cl
```

37. Write a Python program to replace dictionary values with their sum.

ANS=

```
def sum_math_v_vi_average(list_of_dicts):

for d in list_of_dicts:

        n1 = d.pop('V')

n2 = d.pop('VI')

d['V+VI'] = (n1 + n2)/2

return list_of_dicts

student_details= [

  {'id' : 1, 'subject' : 'math', 'V' : 70, 'VI' : 82},

  {'id' : 2, 'subject' : 'math', 'V' : 73, 'VI' : 74},

  {'id' : 3, 'subject' : 'math', 'V' : 75, 'VI' : 86}

]
print(sum_math_v_vi_average(student_details))
```

Output:

```
[{'subject': 'math', 'id': 1, 'V+VI': 76.0}, {'subject': 'math',
'id': 2, 'V+VI': 73.5}, {'subject': 'math', ' id':
3, 'V+VI': 80.5}]
```

38. Write a Python program to match key values in two dictionaries.
Sample dictionary: {'key1': 1, 'key2': 3, 'key3': 2}, {'key1': 1, 'key2': 2}
Expected output: key1: 1 is present in both x and y

ANS=

```
x = {'key1': 1, 'key2': 3, 'key3': 2} y = {'key1':

1, 'key2': 2} for (key, value) in set(x.items()) &

set(y.items()):

    print('%s: %s is present in both x and y' % (key, value))
```

Output: `key1: 1 is present in both x and y`