Name: VIBIN R
Roll No: 20BMC046

KUMARAGURU
COLLEGE OF TECHNOLOGY
LABORATORY WORK BOOK

Exercise/Experiment Number: 9

Lab Code / Lab	:	**U18CSI2201- PYTHON PROGRAMMING LAB**
Course / Branch	:	**I BE /BTech**
Title of the exercise/experiment	:	Implement dictionary and set in python

1. 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}

Code:

```
#Vibin_20BMC046
n=int(input("Enter the no. of terms to be generated:"))
d={}
for i in range(1,n+1):
   d[i]=i*i
print(d)
```

Output:

```
In [1]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter the no. of terms to be generated:8
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
```

Name: VIBIN R
Roll No: 20BMC046

2. Write a Python program to create an intersection of sets.

Code:

```
#Vibin_20BMC046
a,b={1,3,5,7,9,11,25},{1,4,9,16,25}
print("A={0}\nB={1}".format(a,b))
print("A intersection B is ",a&b)
```

Output:

```
In [2]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
A={1, 3, 5, 7, 9, 11, 25}
B={1, 4, 9, 16, 25}
A intersection B is  {1, 9, 25}
```

---

3. Write a Python program to find maximum and the minimum value in a set.

Solution:

```
In [4]: set_a={24,78,14,75,25,8,47,87}

In [5]: max(set_a)
Out[5]: 87

In [6]: min(set_a)
Out[6]: 8
```

4. Create a dictionary with month name and number of days in that month. In this dictionary, keys are month names and whose values are the number of days in the corresponding months.

days = {'January':31, 'February':28, 'March':31, 'April':30, 'May':31, 'June':30, 'July':31, 'August':31, 'September':30, 'October':31, 'November':30, 'December':31}

(a) Print out all of the months with 31 days.
(b) Ask the user to enter a month name and use the dictionary to tell them how many days are in the month.
(c) Print the names of all months in alphabetical order.

Name: VIBIN R
Roll No: 20BMC046

Code:

```
#Vibin_20BMC046
days =
{'January':31,'February':28,'March':31,'April':30,'May':31,'June':30,'July':31,'August':31,'Sept
ember':30,'October':31,'November':30,'December':31}
a=days.items()
print("Months with 31 days:")
for i in a:       #to print months with 31 days
   if i[1]==31:
      print(i[0])
m=input("Enter a month:")
print("{0} has {1} days".format(m,days.get(m)))   #answer for 'b'
print("\nThe Sorted names of Months is\n",sorted(days)) #sorting name of months
```

Output:

```
In [19]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
Months with 31 days:
January
March
May
July
August
October
December

Enter a month:June
June has 30 days

The Sorted names of Months is
 ['April', 'August', 'December', 'February', 'January', 'July', 'June',
'March', 'May', 'November', 'October', 'September']
```

Name: VIBIN R
Roll No: 20BMC046

5. Write a program to check whether a given string is a pangram or not. Use sets.

Code:

```
#Vibin_20BMC046
s=input("Enter a string to check for Pangram:").lower()
a=set('abcdefghijklmnopqrstuvwxyz')
m=a-set(s)
print("Pangram" if(not m) else "Missing Letters:\n{0}".format(m))
```

Output:

```
In [30]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter a string to check for Pangram:Quick fox jumps nightly above wizard
Pangram

In [31]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')

Enter a string to check for Pangram:Welcome to Python Programming
Missing Letters:
{'b', 'f', 'z', 'v', 'x', 'u', 'd', 'q', 'j', 'k', 's'}
```

Name: VIBIN R
Roll No: 20BMC046

KUMARAGURU
COLLEGE OF TECHNOLOGY
LABORATORY WORK BOOK

Exercise/Experiment Number: 10

Lab Code / Lab : **U18CSI2201- PYTHON PROGRAMMING LAB**

Course / Branch : **I BE /BTech**

Title of the exercise/experiment : Working with Tuples.

---

1. Write a Python program to convert a tuple to a string.

Code:

```
#Vibin_20BMC046
a=('Welcome','to','Python','Programming')
print("The Tuple is:\n",a)
s=''
for i in a:
   s=s+' '+i
print("Tuple to String:\n",s)
```

Output:

```
In [1]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
The Tuple is:
 ('Welcome', 'to', 'Python', 'Programming')
Tuple to String:
  Welcome to Python Programming
```

Name: VIBIN R
Roll No: 20BMC046

2. Define a function which can generate and print a tuple where the values are square of numbers between 1 and 20 (both included).

Code:

```
#Vibin_20BMC046
def squares():
   sq=[i*i for i in range(1,21)]
   print("The Squares from 1 to 20:\n",tuple(sq))
squares()
```

Output:

```
In [7]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
The Squares from 1 to 20:
 (1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289,
324, 361, 400)
```

3. Create three lists namely rollno, name and age with necessary values. Write a program to zip the lists and display the student details in the format [('19BEE01', 'Arun', 18), ('19BEE01', 'Preethi', 17), ('19BEE01', 'Shameer', 18)].

Code:

```
#Vibin_20BMC046
Roll=('20BEE036','20BIT014','19BMC046')
name=('Varun','Naveen','Pradeesh')
age=(19,18,21)
print("Student Details:\n",list(zip(Roll,name,age)))
```

Output:

```
In [8]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
Student Details:
 [('20BEE036', 'Varun', 19), ('20BIT014', 'Naveen', 18), ('19BMC046',
'Pradeesh', 21)]
```

Name: VIBIN R
Roll No: 20BMC046

4. Write a Python program to unzip the given list of tuples into individual lists.

Code:

```
#Vibin_20BMC046
L=[("Red", "Circle", 40), ("Blue", "Square", 60), ("Orange", "Rectangle", 90), ("Black",
"Triangle", 25)]
print("The List of Tuples:\n",L)
a,b,c=zip(*L)
print("Unzipped Lists:")
print(list(a),'\n',list(b),'\n',list(c))
```

Output:

```
In [12]: runfile('C:/Users/Vibin/.spyder-py3/temp.py', wdir='C:/Users/
Vibin/.spyder-py3')
The List of Tuples:
 [('Red', 'Circle', 40), ('Blue', 'Square', 60), ('Orange', 'Rectangle', 90),
('Black', 'Triangle', 25)]
Unzipped Lists:
['Red', 'Blue', 'Orange', 'Black']
 ['Circle', 'Square', 'Rectangle', 'Triangle']
 [40, 60, 90, 25]
```