S.No	Programs	Date	Signature
1.	Taking two numbers an input and print the addition.		
2.	Given two numers as input, print the smaller number.		
3.	If else problem of flipping one digit out of 0's and 1's digit to make them same.		
4.	Given a list A of numbers, print those numbers which are multiple of 5		
5.	Given a list of numbers, find the max and 2nd min in the list		
6.	Design a program to take a word as input than encode it into pig latin		
7.	Write a list of numbers and makes a new list of only the first and last elements		
8.	A python script to print a dictionary from No 1-15 and the values are their squares		
9.	Python program to swap two numbers using tuple		
10.	Program to conceptualize multithreading		
11.	Program to display the number of cookies sold on a day by an airport shop		
12.	Write a function to tell user if he/she is able to vote or not		
13.	To print multiplication table of number entered by user using recursion		
14.	To write a function 'perfect' to determine if a number is perfect number		
15.	To create a file student and write 5 students name on file and read from file		
16.	To check whether lowercase letters exist in a string		

17.	Python program to replace a white space with an underscore and vice versa		
18.	Program to find all the three, four, five characters long words in a string		
19.	Program to show exception handling		
20.	Program to draw rainbow benzene		
21.	Program to draw multiple squares all starting from a common point		
22.	Program to draw an Image Graphic		
23.	Program for a Gray Scale		
24.	Python program to demonstrate Inheritance		

1.Program to take two integer number as input and print the addition.

INPUT:

```
x=int(input("Enter first number:"))
y=int(input("Enter second number:"))
print(x,y,sep=" ")
print(x+y)
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/practice6.py
Enter first number:4
Enter second number:2
4 2
6
```

2.Given two numbers(integers) as input, print the smaller number.

INPUT:

```
x=int(input("Enter first number:"))
y=int(input("Enter second number:"))
print(x,y,sep=" ")
if x>y:
    print(y) #prints the smaller number
else:
    print(x)
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/Q2.py
Enter first number:2
Enter second number:3
2 3
2
```

3.Your task is to make all digits same by just flipping one digit only.
If it is possible to make all the digits same by just flipping one digit then Print ‘Yes’ else print ‘No’

INPUT :

```
str=input("Enter number:")
def can_make_all_same(str):
    zeros=0
    ones=0
    for i in range(0,len(str)):
        ch=str[i];
        if (ch=='0'):
            zeros=zeros+1
        else:
            ones=ones+1
        return(zeros==1 or ones==1);
if can_make_all_same(str):
    print("Yes")
else:
    print("No")
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/4th.py
Enter number:11
Yes
PS D:\python>
```

4.Given a list A of numbers(integers), you have to print those numbers which are not a multiple of 5.

INPUT :

```
n=int(input("the length of list:"))
number=[]
for i in range(1,n+1):
    i=int(input("Enter list items:"))
    number.append(i)

print(number)
for i in number[:n]:
    if i%5==0:
     number.remove(i)
     print(number)
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/Q3.py
the length of list:6
Enter list items:1
Enter list items:2
Enter list items:3
Enter list items:4
Enter list items:5
Enter list items:6
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 6]
```

5.Given a list of numbers(integers) , find the second maximum and second minimum in the list.

INPUT :

```
n=int(input("the length of list:"))
number=[]
for i in range(1,n+1):
    i=int(input("Enter list items:"))
    number.append(i)
print(number)
number.sort()
print(number[-2]) #second maximum
print(number[1])  #second minimum
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/Q6.py
the length of list:5
Enter list items:1
Enter list items:2
Enter list items:3
Enter list items:4
Enter list items:5
[1, 2, 3, 4, 5]
4
2
```

6.Design a program to take a word as input, and then encode it into pig-latins.

INPUT:

```
vowels=('A','E','I','O','U')
consonants=('B','C','D','F','G','H','J','K','L','M','N','O','P','Q','R'
,'S','T','V','W','X','Y','Z')
st=input("Enter Word:")
first=st[0]
first=str(first)
first=first.upper()
if first in consonants:
    remove_first=st[1:]
    pig_latin=remove_first+first+'AY'
    print(pig_latin)
    if first in vowels:
        pig_latin=st+'WAY'
        print(pig_latin)
```

OUTPUT:

```
PS C:\Users\rohil\OneDrive\Documents\Python Tutorials> python -u "c:\Users\rohil\OneDrive\Documents\Python Tutorials\1.py"
Enter Word:PARIS
ARISPAY
```

7.Write a program that takes a list of numbers and makes a new list of only the first and last elements of the given list.

INPUT:

```
n=int(input("the length of list:"))
number=[]
for i in range(1,n+1):
    i=int(input("Enter list items:"))
    number.append(i)
print(number)
num2=['a','b']
num2[0]=number[0]
num2[1]=number[-1]
print(num2)
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/lower.py
the length of list:5
Enter list items:5
Enter list items:10
Enter list items:15
Enter list items:20
Enter list items:25
[5, 10, 15, 20, 25]
[5, 25]
```

8.Write a python script to print a dictionary where the keys are numbers from 1 to 15 and the values are their squares.

INPUT:

```
squares={"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"}

print(squares)
```

Output:

```
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/lower.py
{'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',
196', '15': '225'}
PS D:\python>
```

9.Write a python program to swap two numbers using tuple.

INPUT:

```
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
print("a=",a, "b=",b)
(a,b)=(b,a)
print("a=",a, "b=",b)
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/lower.py
Enter first number:5
Enter second number:2
a= 5 b= 2
a= 2 b= 5
PS D:\python>
```

10.Program to conceptualize multithreading

INPUT

```
import threading
import OS
def task1():
    print("Task  1  assigned  to  thread:
{}".format(threading.current_thread().na
me))
    print("ID of process running task 1: {}".format(os.getpid()))
   def task2():
       print("Task 2 assigned to thread:
{}".format(threading.current_thread().na
me))
    print("ID of process running task 2: {}".format(os.getpid()))
   if __name__ ==
"__main__":
    # print ID of current
process
    print("ID of process running main program:
{}".format(os.getpid()))
    # print name of main thread
    print("Main thread name:
{}".format(threading.current_thread().name))
    # creating threads
    t1 = threading.Thread(target=task1,
name='t1')
    t2 =
threading.Thread(target=task2,
name='t2')
    # starting
threads
t1.start()
t2.start()
    # wait until all
threads finish
t1.join()
t2.join()
```

output :

```
ID of process running main program: 11758
Main thread name: Main Thread
Task 1 assigned to thread: t1
ID of process running task 1:
11758 Task 2 assigned to
thread: t2
ID of process running task 2: 11758
```

11.Write a program using function to find and display the number of cookies sold on a day by a shop in the airport. Assume that the flight which took off are filled to the capacity, flights which landed are half filled and all passengers bought 2 cookies each.

INPUT:

```
F1=int(input("Enter no. of flights that took off:"))
F2=int(input("Enter no. of flights that landed:"))
P1=int(input("No. of passengers in flights that took off:"))
P2=int(input("No. of passengers in flights that landed:"))
Cookies_sold=(F1*P1*2)+((F2/2)*P2*2)
print(Cookies_sold)
```

Output:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/d.py
Enter no. of flights that took off:100
Enter no. of flights that landed:110
No. of passengers in flights that took off:150
No. of passengers in flights that landed:185
50350.0
```

12.Write a function to tell user if he/she is able to vote or not.

INPUT:

```python
def voting():
    age=int(input("Enter your age:"))
    if age>=18:
        print("You are eligible to vote!")
    else:
        print("Not eligible to vote!")
voting()
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/2nd.py
Enter your age:19
You are eligible to vote!
PS D:\python>
```

13.Print multiplication table of number entered by user using recursion.

INPUT:

```python
n=int(input("Enter number:"))
def table(n):
    for i in range(1,11):
        print(n*i)
table(n)
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/3rd.py
Enter number:4
4
8
12
16
20
24
28
32
36
40
```

14.Write a function “perfect()” that determines if parameter number is a perfect number .Use this function in a program that determines and prints all the perfect number till 1000.

INPUT:

```
def Perfect( n ):
    sum = 1
    i = 2
    while i * i <= n:
        if n % i == 0:
            sum = sum + i + n/i
        i += 1

    return (True if sum == n and n!=1 else False)
print("Below are all perfect numbers till 10000")
n = 2
for n in range (1000):
    if Perfect (n):
        print(n , " is a perfect number")
```

Output:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/practice6.py
Below are all perfect numbers till 10000
6  is a perfect number
28  is a perfect number
496  is a perfect number
```

15.Create a file student and write 5 student name on file and read from file.

Input:

```
f=open("student.txt")
print(f.read())
```

Output:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/file.py
Azad
Ayush
Abhishek
Harsh
Prashant
```

16.Implement a python program to check whether lowercase letters exist in a string.

INPUT:

```
a=input("Enter a string:")
for char in a:
    k=char.islower()
    if k==True:
        print('Yes lowercase letter exist in a string')
        break
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/lower.py
Enter a string:This is a string
Yes lowercase letter exist in a string
PS D:\python>
```

17.Write a python program to replace a whitespaces with an underscore and vice versa.

INPUT:

```
import re
string=input("Enter string:")
string=string.replace(" ", "_")
print(string)
string=string.replace("_"," ")
print(string)
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/lower.py
Enter string:This is a string
This_is_a_string
This is a string
PS D:\python>
```

18.Write a python program to find all the three, four, five characters long words in a string.

INPUT:

```
import re

string=input("Enter string:")
print(re.findall(r"\b\w{3,5}\b", string))
```

OUTPUT:

```
PS D:\python> & C:/Users/Azad/AppData/Local/Programs/Python/Python39/python.exe d:/python/Assignment/lower.py
Enter string:The quick brown fox jumps over the lazy dog
['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']
```

19.Write a program to show exception handling.

INPUT:

```
try:
    print(x)
except:
    print("An exception occurred")
```

OUTPUT:

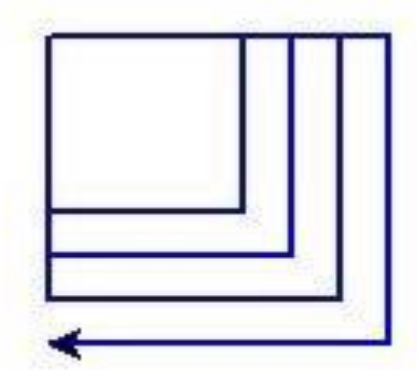
```
PS C:\Users\rohil\OneDrive\Documents\Python Tutorials> python -u '
An exception occurred
```

21.Program to draw multiple squares all starting from a common point.

INPUT :

```
import turtle
mult_square=turtle.Turtle()
def Multiple_Squares(length,
colour):
mult_square.pencolor(colour)
mult_square.pensize(2)
mult_square.forward(length)
mult_square.right(90)
mult_square.forward(length)
mult_square.right(90)
mult_square.forward(length)
mult_square.right(90)
mult_square.forward(length)
mult_square.right(90)
mult_square.setheading(360)
for i in range(60,120,15):
Multiple_Squares(i,"blue")
turtle.done
```

OUTPUT:



23.Program for a Gray scale.

INPUT:

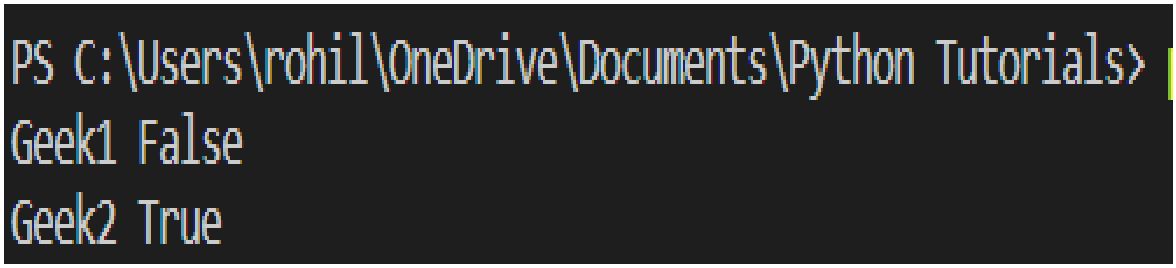
```
# import opencv
import cv2

# Load the input image
image = cv2.imread('C:\\Users\\rohil\\Downloads\\tomato.jpeg')
cv2.imshow('Original', image)
cv2.waitKey(0)

# Use the cvtColor() function to grayscale the image
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

cv2.imshow('Grayscale', gray_image)
cv2.waitKey(0)

# Window shown waits for any key pressing event
cv2.destroyAllWindows()
```

24.Python Program to demostrate Inheritance

INPUT:

```
# Base or Super class. Note object in bracket.
# (Generally, object is made ancestor of all classes)
# In Python 3.x "class Person" is
# equivalent to "class Person(object)"


class Person(object):

        # Constructor
        def __init__(self, name):
                self.name = name

        # To get name
        def getName(self):
                return self.name

        # To check if this person is an employee
        def isEmployee(self):
                return False


# Inherited or Subclass (Note Person in bracket)
class Employee(Person):

        # Here we return true
        def isEmployee(self):
                return True


# Driver code
emp = Person("Geek1") # An Object of Person
print(emp.getName(), emp.isEmployee())

emp = Employee("Geek2") # An Object of Employee
print(emp.getName(), emp.isEmployee())
```

OUTPUT:


PS C:\Users\rohil\OneDrive\Documents\Python Tutorials>
Geek1 False
Geek2 True