Jamia Millia Islamia

Dept. of Computer Science

Subject: Pattern Matching Using Python

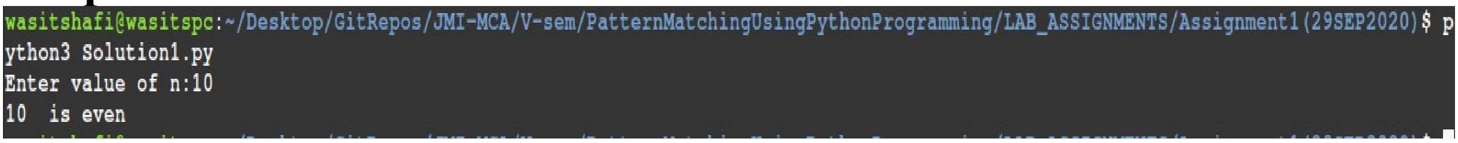
Submitted By: Wasit Shafi **Submitted To:**Dr.Taran Singh Bharti
Roll No: 18MCA054
Date: 11-OCT-2020

Q1. Write a program to perform input and output operations.

Sol.

```
n = int(input('Enter value of n:')) # Input Operation
if n % 2 == 0:
  print(n ,' is even') # Output Operation
else:
  print(n, ' is odd')
```

Output:



```
wasitshafi@wasitspc:~/Desktop/GitRepos/JMI-MCA/V-sem/PatternMatchingUsingPythonProgramming/LAB_ASSIGNMENTS/Assignment1(29SEP2020)$ p
ython3 Solution1.py
Enter value of n:10
10  is even
```

Q2. Write a programs to illustrate decision making statements i.e. displaying division of students based on their marks, prime number, leap year, searcing, arranging values.

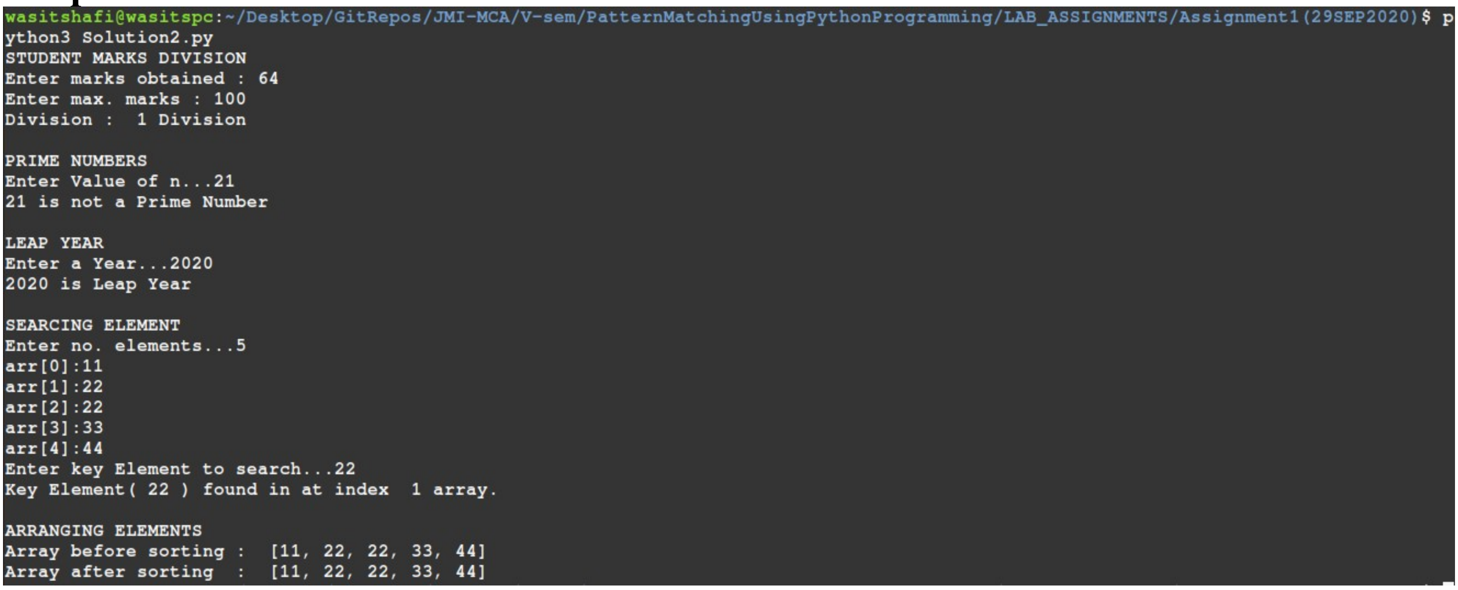
Sol.

```
def get_division(p):
   if p >= 75:
     return "Distiction"
   elif p >=60:
     return "1 Division"
   elif p >=50:
     return "2 Division"
   elif p >=40:
     return "3 Division"
   else:
     return "4 Division(Fail)"

def isPrime(n):
  if n < 2:
    return False;
  else:
    i = 2
    while i * i <= n:
      if n % i == 0:
        return False
      i += 1
    return True

def isLeapYear(year):
  return True if ((year % 4 == 0 and year % 100 != 0) or
(year % 400 == 0)) else False
```

```
def searchKey(arr, key):
  length = len(arr)
  for i in range(length):
    if arr[i] == key:
      return i
  return -1;

def mySort(arr): # Using Bubble Sort Algorithm
  n = len(arr)
  for i in range(n - 1):
    for j in range(n - i - 1):
      if arr[j] > arr[j + 1]:
        temp = arr[j]
        arr[j] = arr[j + 1]
        arr[j +1] = temp
# Division Of Marks
print("STUDENT MARKS DIVISION")
marksObtained = int(input('Enter marks obtained : '))
maxMarks = int(input('Enter max. marks : '));
percentage = (marksObtained / maxMarks) * 100
print('Division : ', get_division(percentage))

# Prime Numbers
print("\nPRIME NUMBERS")
n = int(input('Enter Value of n...'))
if isPrime(n):
  print(n , 'is Prime Number')
else:
  print(n, 'is not a Prime Number')

# Leap Year
print("\nLEAP YEAR")
year = int(input('Enter a Year...'))
if isLeapYear(year):
  print(year , 'is Leap Year')
else:
  print(year, 'is not a Leap Year')

# Searching
print("\nSEARCING ELEMENT")
n = int(input('Enter no. elements...'))
arr = [None] * n
for i in range(n):
  arr[i] = int(input('arr[' + str(i) +']:'))
key = int(input('Enter key Element to search...'))
```

```
index = searchKey(arr, key)
if index == -1:
  print('Key Element(', key, ') not found in array.')
else:
  print('Key Element(', key, ') found in at index ', index,
'array.')

# Arranging
print("\nARRANGING ELEMENTS")
print("Array before sorting : ", arr)
mySort(arr) # soring in non-decreasing order
print("Array after sorting  : ", arr)
```

Output :



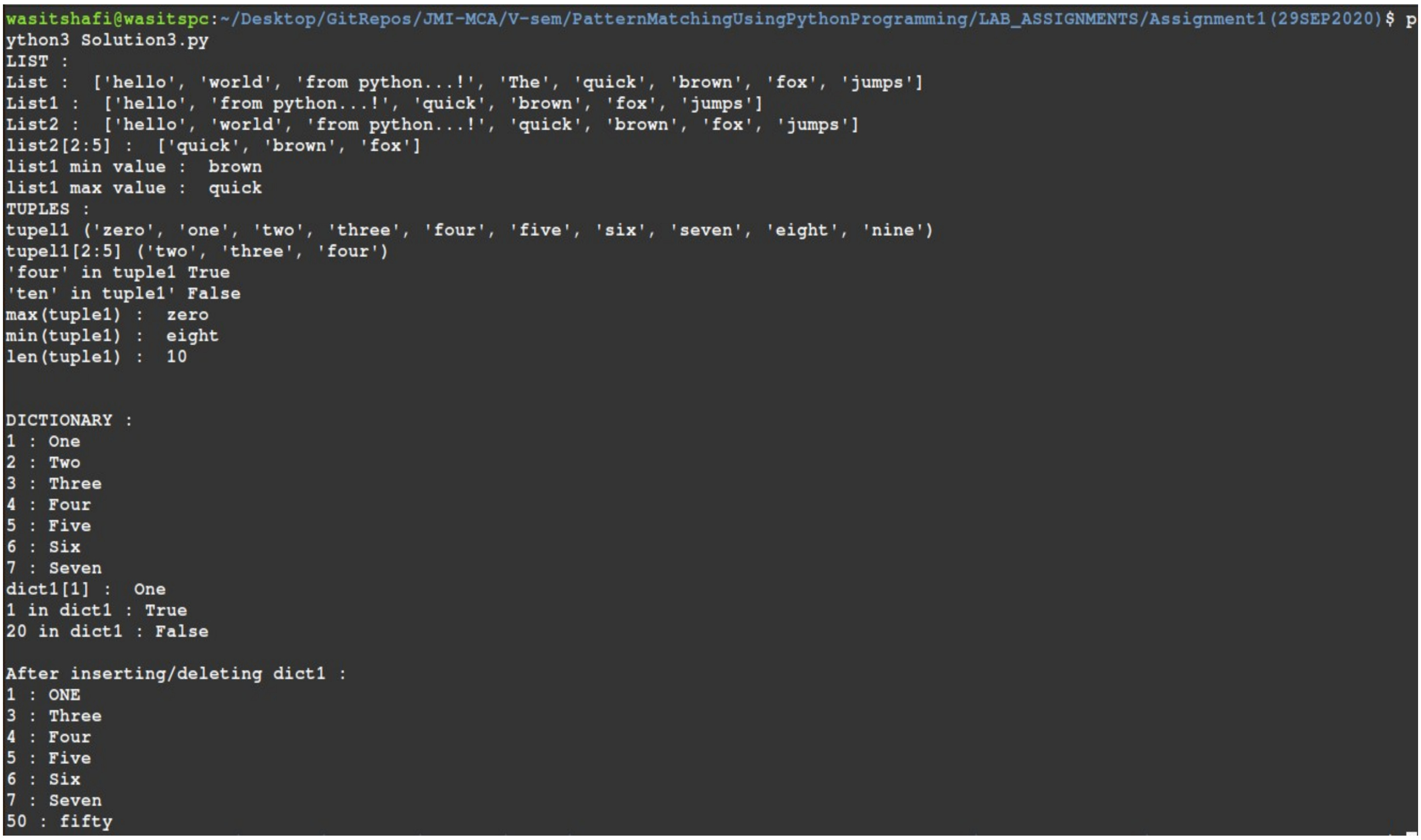
```
wasitshafi@wasitspc:~/Desktop/GitRepos/JMI-MCA/V-sem/PatternMatchingUsingPythonProgramming/LAB_ASSIGNMENTS/Assignment1(29SEP2020)$ p
ython3 Solution2.py
STUDENT MARKS DIVISION
Enter marks obtained : 64
Enter max. marks : 100
Division :  1 Division

PRIME NUMBERS
Enter Value of n...21
21 is not a Prime Number

LEAP YEAR
Enter a Year...2020
2020 is Leap Year

SEARCING ELEMENT
Enter no. elements...5
arr[0]:11
arr[1]:22
arr[2]:22
arr[3]:33
arr[4]:44
Enter key Element to search...22
Key Element( 22 ) found in at index  1 array.

ARRANGING ELEMENTS
Array before sorting :  [11, 22, 22, 33, 44]
Array after sorting  :  [11, 22, 22, 33, 44]
```

Q3. Write a program illustrating the functioning (creation, inserting, deletion, max, min etc) of list, tuples and dictionary.

Sol.

```
# List : List is a collection which is ordered and
changeable. Allows duplicate members.
print("LIST :")
list1 = ['hello', 'world', 'The', 'quick', 'brown']
list1.append('fox') # added at end of the list
list1.append('jumps')
list1.insert(2,'from python...!') # added at specific index
print('List : ', list1)
list1.remove('The')       # deleting element by value
list2 = list1.copy()      # copy list1 elements to list2
del list1[1]              # deleting element by index
print('List1 : ', list1)
print('List2 : ', list2)
print('list2[2:5] : ', list1[2:5])
print('list1 min value : ', min(list1)) # lexicographically
smallest
print('list1 max value : ', max(list1))



# Tuples : A tuple is a collection which is ordered and
unchangeable. In Python tuples are written with round
brackets.
print("TUPLES :")
tuple1 = ('zero', 'one', 'two', 'three', 'four', 'five',
'six', 'seven', 'eight', 'nine')
print('tupel1', tuple1)
print('tupel1[2:5]', tuple1[2:5])
print("'four' in tuple1", 'four' in tuple1 )
print("'ten' in tuple1'", 'ten' in tuple1 )
print('max(tuple1) : ' , max(tuple1))
print('min(tuple1) : ' , min(tuple1))
print('len(tuple1) : ', len(tuple1))



# Dictionary : Dictionary is a collection which is
unordered, changeable and indexed. No duplicate members.
print("\n\nDICTIONARY :")
dict1 = {1 : 'One', 2 : 'Two', 3 : 'Three', 4 : 'Four', 5 :
'Five', 6 : 'Six', 7 : 'Seven'}

for key, value in dict1.items():   # printing Dictionary
  print(key, ':', value)

print('dict1[1] : ', dict1[1])     # return value of key
'1', if key is not found then it raises an error
print("1 in dict1 :", 1 in dict1)  # return true if dict1
contains key 1, else return false
print("20 in dict1 :", 20 in dict1)# return true if dict1
contains key 1, else return false

dict1.pop(2)       # removes element
dict1[1] = 'ONE'   # if dict1 contains key, then it
overwrites value else it insert as a new element
dict1[50] = 'fifty'
print("\nAfter inserting/deleting dict1 : ")
for key, value in dict1.items():  # printing Dictionary
  print(key, ':', value)
```

Output



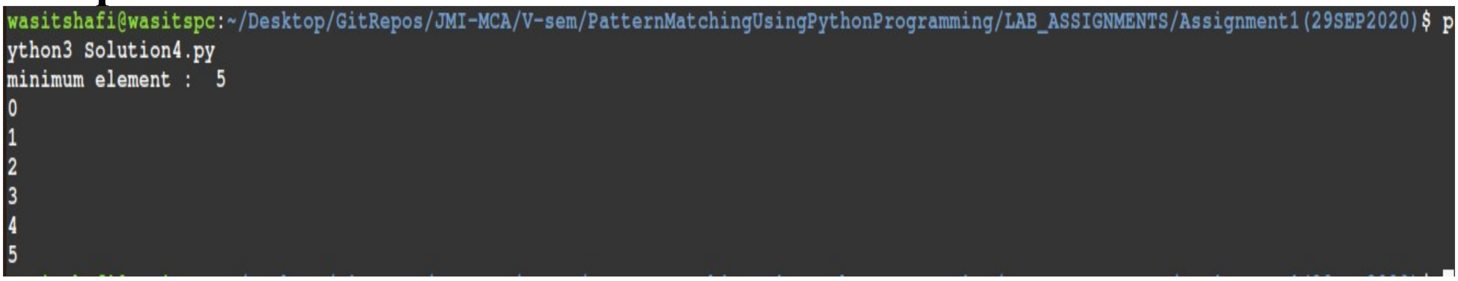
```
wasitshafi@wasitspc:~/Desktop/GitRepos/JMI-MCA/V-sem/PatternMatchingUsingPythonProgramming/LAB_ASSIGNMENTS/Assignment1(29SEP2020)$ p
ython3 Solution3.py
LIST :
List :  ['hello', 'world', 'from python...!', 'The', 'quick', 'brown', 'fox', 'jumps']
List1 :  ['hello', 'from python...!', 'quick', 'brown', 'fox', 'jumps']
List2 :  ['hello', 'world', 'from python...!', 'quick', 'brown', 'fox', 'jumps']
list2[2:5] :  ['quick', 'brown', 'fox']
list1 min value :  brown
list1 max value :  quick
TUPLES :
tupel1 ('zero', 'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine')
tupel1[2:5] ('two', 'three', 'four')
'four' in tuple1 True
'ten' in tuple1' False
max(tuple1) :  zero
min(tuple1) :  eight
len(tuple1) :  10


DICTIONARY :
1 : One
2 : Two
3 : Three
4 : Four
5 : Five
6 : Six
7 : Seven
dict1[1] :  One
1 in dict1 : True
20 in dict1 : False

After inserting/deleting dict1 :
1 : ONE
3 : Three
4 : Four
5 : Five
6 : Six
7 : Seven
50 : fifty
```

Q4. write a program using functions and procedures.

Sol.

```
def func1(x, y): # function to return minimum
  if x < y:
    return x
  else:
    return y

def proc1(n): # procedure to print 0 - n
  for i in range(n+1):
    print(i)

minimum = func1(10, 5)
print('minimum element : ', minimum)
proc1(minimum)
```

Output



```
wasitshafi@wasitspc:~/Desktop/GitRepos/JMI-MCA/V-sem/PatternMatchingUsingPythonProgramming/LAB_ASSIGNMENTS/Assignment1(29SEP2020)$ p
ython3 Solution4.py
minimum element :  5
0
1
2
3
4
5
```

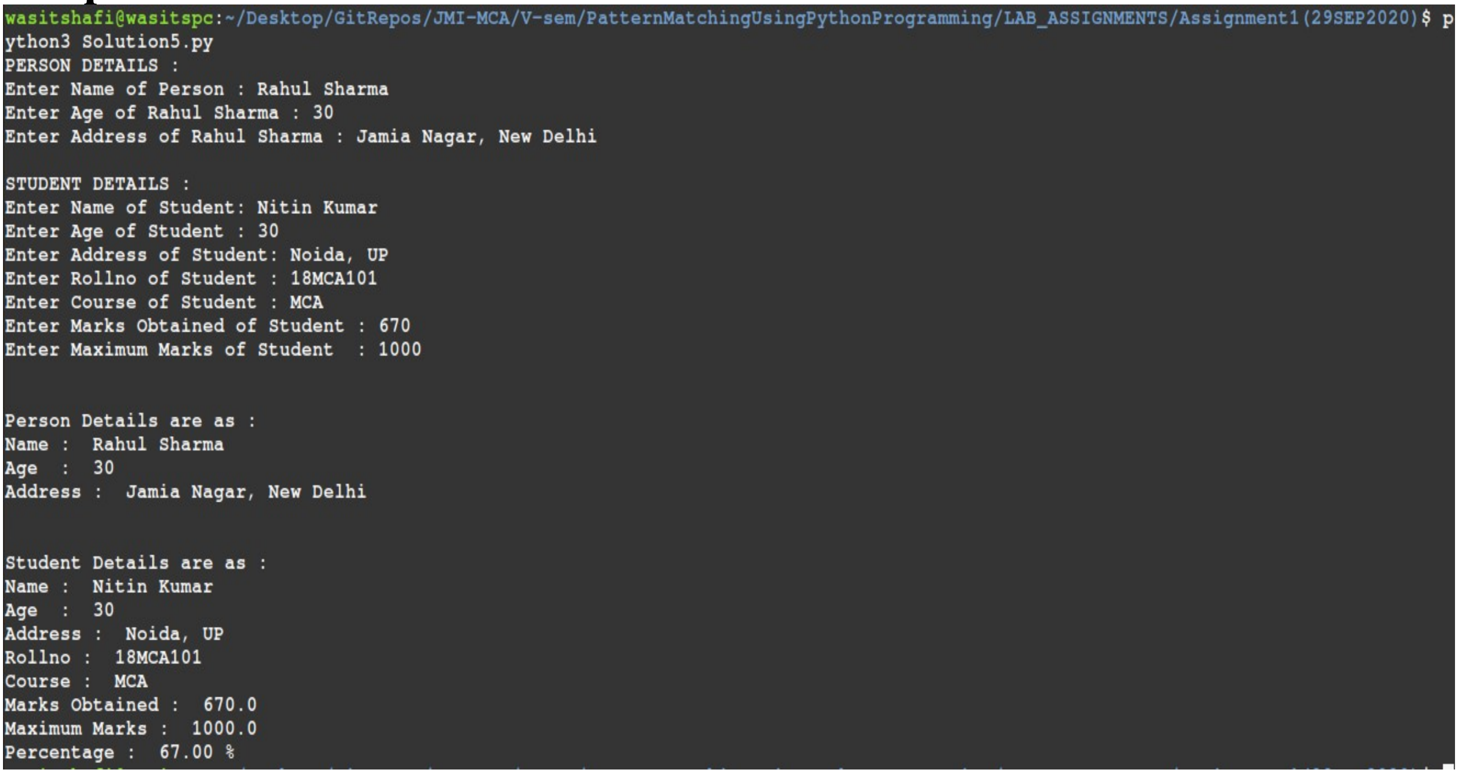
Q5. Write a program to implement OOPs concepts in python i.e. inheritance etc.

Sol.

```
class Person:
  def __init__(self, name, age, address): # constructor
    self.name = name
    self.age = age
    self.address = address

  def set_data(self, name, age, address):
    self.name = name
    self.age = age
    self.address = address

  def get_data(self):
    print('Name : ', self.name)
    print('Age  : ', self.age)
    print('Address : ', self.address)

  def get_name(self):
    return self.name

  def get_age(self):
    return self.age

  def get_address(self):
    return self.address

class Student(Person):
  def __init__(self, name, age, address, rollno, course, 
marksObtained, maximumMarks): # constructor
    super().__init__(name, age, address)
    self.rollno = rollno
    self.course = course
    self.marksObtained = marksObtained
    self.maximumMarks = maximumMarks

  def set_data(self, rollno, course, marksObtained, 
maximumMarks):
    self.rollno = rollno
    self.course = course
    self.marksObtained = marksObtained
    self.maximumMarks = maximumMarks

  def get_data(self):
    print('Name : ', super().get_name())
    print('Age  : ', super().get_age())
    print('Address : ', super().get_address())
    print('Rollno : ', self.rollno)
    print('Course : ', self.course)
    print('Marks Obtained : ', self.marksObtained)
    print('Maximum Marks : ', self.maximumMarks)
    print('Percentage : ',
'{:.2f}'.format(self.get_percentage()), '%')

  def get_rollno(self):
    return self.rollno

  def get_course(self):
    return self.course

  def get_marksObtained(self):
    return self.marksObtained

  def maximumMarks(self):
    return self.maximumMarks

  def get_percentage(self):
    return marksObtained / maximumMarks * 100

print('PERSON DETAILS : ')
name = input('Enter Name of Person : ')
age = int(input('Enter Age of ' + name + ' : '))
address = input('Enter Address of ' + name +' : ')
p1 = Person(name, age, address) # Object of Person Class

print('\nSTUDENT DETAILS : ')
name = input('Enter Name of Student: ')
age = int(input('Enter Age of Student : '))
address = input('Enter Address of Student: ')
rollno = input('Enter Rollno of Student : ')
course = input('Enter Course of Student : ')
marksObtained = float(input('Enter Marks Obtained of Student
: '))
maximumMarks = float(input('Enter Maximum Marks of
Student  : '))
s1 = Student(name, age, address, rollno, course,
marksObtained, maximumMarks)

# Printing Person Details
print("\n\nPerson Details are as :")
p1.get_data();
```

```
#Printing Student Detatils
print("\n\nStudent Details are as :")
s1.get_data();
```

Output



```
wasitshafi@wasitspc:~/Desktop/GitRepos/JMI-MCA/V-sem/PatternMatchingUsingPythonProgramming/LAB_ASSIGNMENTS/Assignment1(29SEP2020)$ p
ython3 Solution5.py
PERSON DETAILS :
Enter Name of Person : Rahul Sharma
Enter Age of Rahul Sharma : 30
Enter Address of Rahul Sharma : Jamia Nagar, New Delhi

STUDENT DETAILS :
Enter Name of Student: Nitin Kumar
Enter Age of Student : 30
Enter Address of Student: Noida, UP
Enter Rollno of Student : 18MCA101
Enter Course of Student : MCA
Enter Marks Obtained of Student : 670
Enter Maximum Marks of Student  : 1000


Person Details are as :
Name :  Rahul Sharma
Age  :  30
Address :  Jamia Nagar, New Delhi


Student Details are as :
Name :  Nitin Kumar
Age  :  30
Address :  Noida, UP
Rollno :  18MCA101
Course :  MCA
Marks Obtained :  670.0
Maximum Marks :  1000.0
Percentage :  67.00 %
```

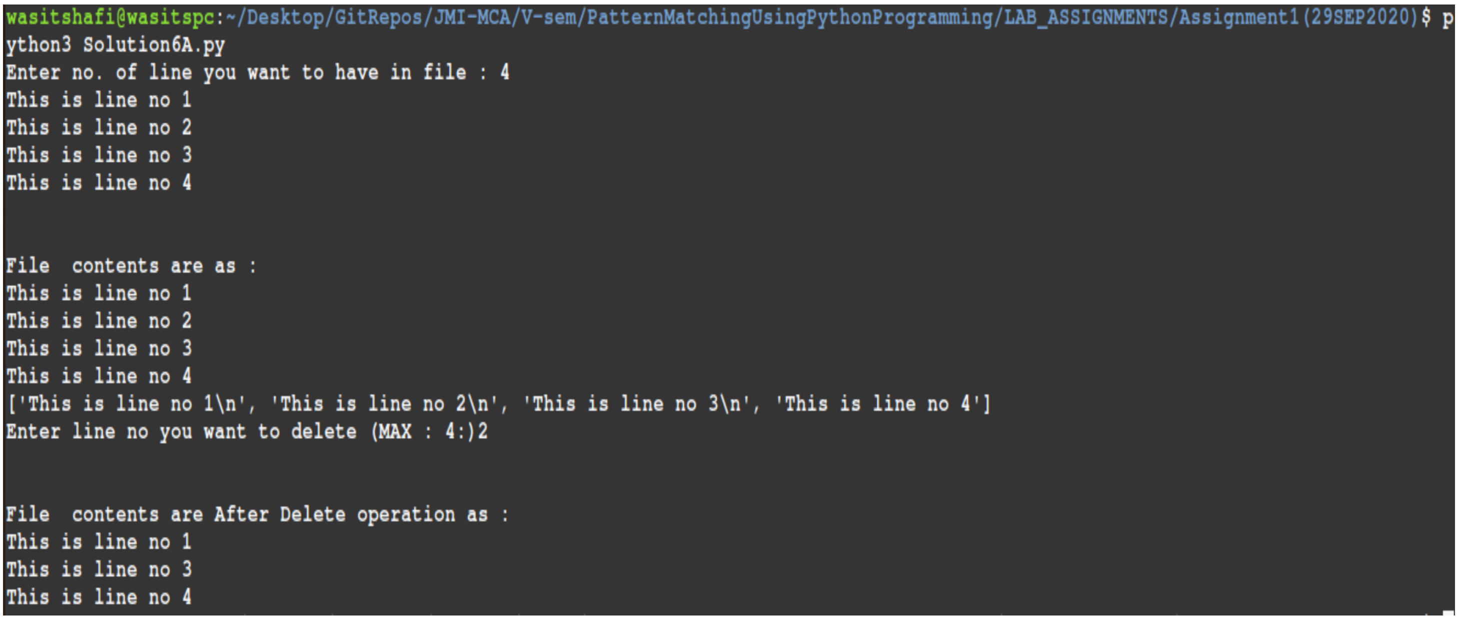
Q6. Write a program for creation, insertion, deletion operation in files, text files, csv files, excel files, etc.

Sol.

Text File Operations

```
def read_file(file):
  for line in file.readlines():
    print(line.rstrip('\n'))

def write_file(file):
  n = int(input('Enter no. of line you want to have in 
file : '))

  for i in range(n):
    file.write(input())
    if i != n-1 :
      file.write("\n")

def delete_line(file_name):
  file = open(file_name, 'r')
  lines = file.readlines()
  print(lines)
  file.close()
  n = int(input('Enter line no you want to delete (MAX : ' +
str(len(lines)) + ':)'))
  if n > len(lines):
    print('Invalid Line no.')
    return

  file = open(file_name, 'w')
  for i in range(len(lines)):
    if i+1 != n:
      if i + 2 == n and i + 2 == len(lines): # if we have to
delete the last line
        file.write(lines[i].rstrip('\n'))
      else:
        file.write(lines[i])
  file.close()

### Main ###
# Creating new text file
file_name = 'file1.txt'
file = open(file_name, 'w') # opening a file in write mode

# Writing file
write_file(file)
file.close()

file = open(file_name, 'r+') # opening a file in read &
write mode
# Read file
#print("File  contents are as : ", file.read())  # Method 1
# Method 2
print("\n\nFile  contents are as : ")
read_file(file)
file.close()

# Deleting a specific line from text file
#delete_line(file_name)
delete_line('file1.txt')
print("\n\nFile  contents are After Delete operation as : ")
file = open(file_name, 'r+') # opening a file in read &
write mode
read_file(file)
file.close()
```

Output



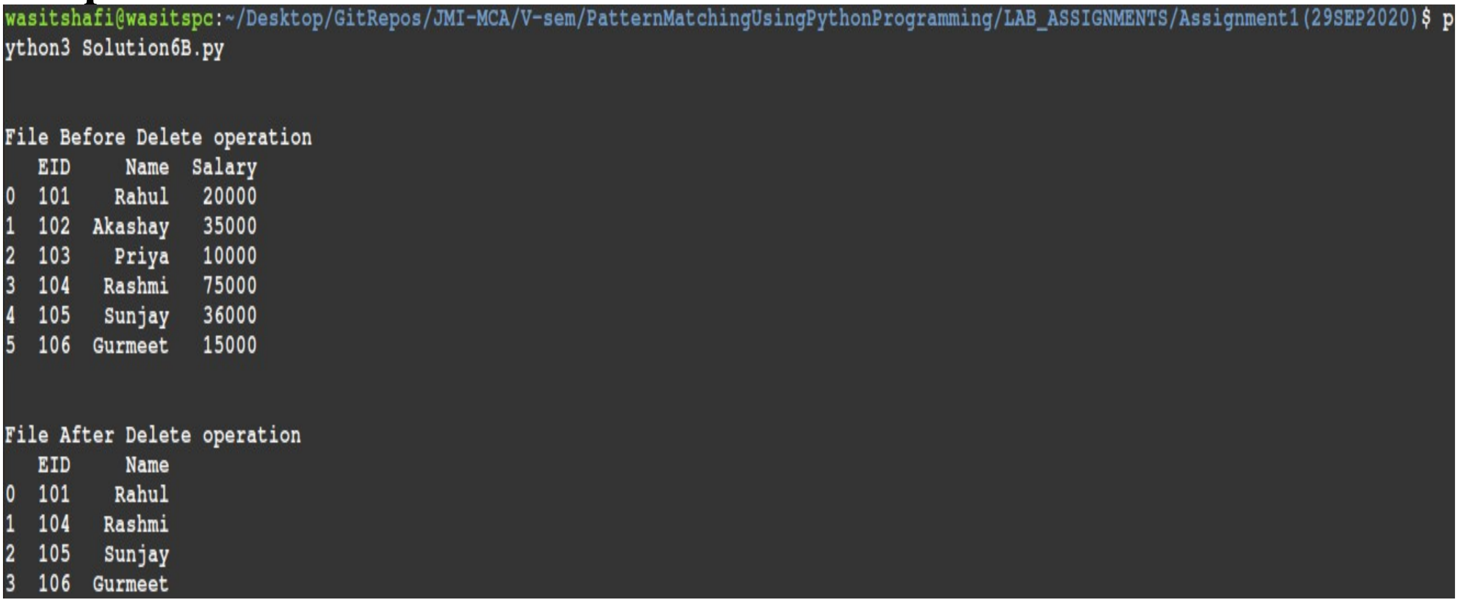
```
wasitshafi@wasitspc:~/Desktop/GitRepos/JMI-MCA/V-sem/PatternMatchingUsingPythonProgramming/LAB_ASSIGNMENTS/Assignment1(29SEP2020)$ p
ython3 Solution6A.py
Enter no. of line you want to have in file : 4
This is line no 1
This is line no 2
This is line no 3
This is line no 4


File  contents are as :
This is line no 1
This is line no 2
This is line no 3
This is line no 4
['This is line no 1\n', 'This is line no 2\n', 'This is line no 3\n', 'This is line no 4']
Enter line no you want to delete (MAX : 4:)2


File  contents are After Delete operation as :
This is line no 1
This is line no 3
This is line no 4
```

CSV File Operations

```
import pandas as pd
# Q6. Write a program for creation, insertion, deletion
operation in files, text files, csv files, excel files, etc.

def write_file(file_name):
  data = {
    'EID'    : [101, 102, 103, 104, 105, 106],
    'Name'   : ['Rahul', 'Akashay', 'Priya' ,'Rashmi',
'Sunjay', 'Gurmeet'],
    'Salary': [20000, 35000, 10000, 75000, 36000, 15000]
  }
  df = pd.DataFrame(data)
  df.to_csv(file_name, index=False)

def read_file(file_name):
  df = pd.read_csv(file_name)
  print(df)

def delete_file(file_name):
  data = pd.read_csv(file_name)
  data.drop([2], inplace = True) # deleting row
  data.drop([1], inplace = True) # deleting row
  data.drop(['Salary'], axis=1, inplace = True) # deleting
col
  data.to_csv(file_name, index = False)
```

```
file_name = 'data1.csv'
# Writing file
write_file(file_name)

# Reading file
print ("\n\nFile Before Delete operation")
read_file(file_name)

# Deleting
delete_file(file_name)
print ("\n\nFile After Delete operation")
read_file(file_name)
```

Output



```
wasitshafi@wasitspc:~/Desktop/GitRepos/JMI-MCA/V-sem/PatternMatchingUsingPythonProgramming/LAB_ASSIGNMENTS/Assignment1(29SEP2020)$ p
ython3 Solution6B.py


File Before Delete operation
   EID     Name  Salary
0  101    Rahul   20000
1  102  Akashay   35000
2  103    Priya   10000
3  104   Rashmi   75000
4  105   Sunjay   36000
5  106  Gurmeet   15000


File After Delete operation
   EID     Name
0  101    Rahul
1  104   Rashmi
2  105   Sunjay
3  106  Gurmeet
```

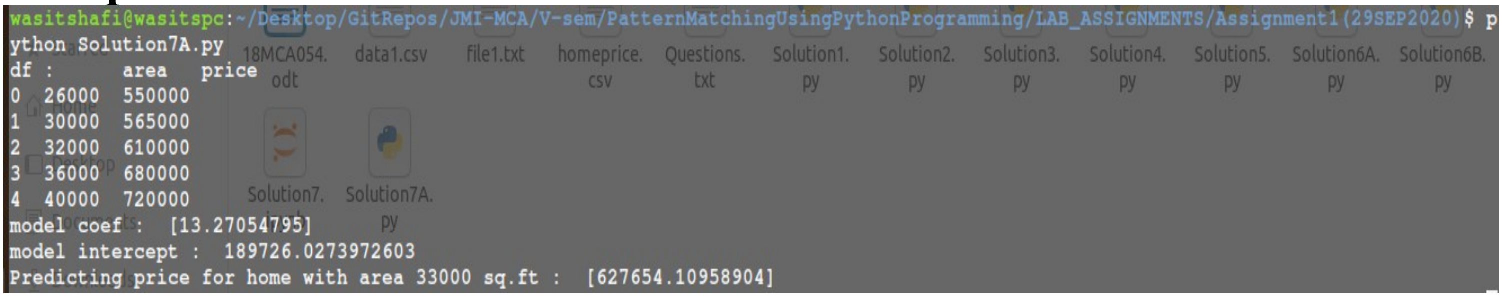
Q7. write a program to perform some data analytics using machine learning packages, i.e. pandas, preprocessing regression, classification, clustering, datasets, etc.

Sol.

Linear Regression

```
import pandas as pd
from sklearn import linear_model

FILE_NAME = 'homeprice.csv'
df = pd.read_csv(FILE_NAME)

print('df : ', df)
model = linear_model.LinearRegression()
model.fit(df[['area']], df['price'])
print('model coef : ', model.coef_)
print('model intercept : ', model.intercept_)
print('Predicting price for home with area 33000 sq.ft : ',
model.predict([[33000]]))
```

Output



```
wasitshafi@wasitspc:~/Desktop/GitRepos/JMI-MCA/V-sem/PatternMatchingUsingPythonProgramming/LAB_ASSIGNMENTS/Assignment1(29SEP2020)$ python Solution7A.py
df :       area   price
0  26000  550000
1  30000  565000
2  32000  610000
3  36000  680000
4  40000  720000
model coef :  [13.27054795]
model intercept :  189726.0273972603
Predicting price for home with area 33000 sq.ft :  [627654.10958904]
```