PRACTICAL FILE

Python Programming (MCA-166)

Submitted by:

Hitesh Walia
Roll no.: 00311104422
Sem: MCA 2nd
Batch: 2022-24

Submitted to:

Ms. Smriti Sharma
Asstt. Professor

Question1

Write a program to find

a. **Sum of Digits of a number**
b. **Product of digits**

Code + Output:

```
# Q1(a) Sum of digits
  # (b) Product of digits
sum=0
product=1
n=int(input("Enter any number: "))
while (n>0):
    i=n%10
    sum=sum + i
    product=product * i
    n=n//10

print("Sum of digits is:",sum)
print("Product of digits is:",product)
```

```
Enter any number: 765
Sum of digits is: 18
Product of digits is: 210
```

Question 2

Write a program for

a. **Multiplication**

Code + Output:

```
# Q2(a) Multiplication
n=int(input("Enter any number: "))
print("Multiplication Table of",n,"is: ")
for i in range(1,11):
  print(n,"X",i,"=",n*i)
```

```
Enter any number: 5
Multiplication Table of 5 is:
5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50
```

b. **Factorial**

Code + Output:

```
# Q2 (b) Factorial
def fact(n):
  if(n==1 or n==0): return n
  return n * fact(n-1)

n=int(input("Enter any number: "))
print("Factorial is:",fact(n))
```

```
Enter any number: 5
Factorial is: 120
```

Question 3

Write a program:

a. **To find numbers divisible by 7 and are not multiple of 5. In range of 2000-3200 inclusive.**

Code + Output:

```
[ ] # Q3(a)
    l=[]
    for i in range(2000, 3201):
      if (i%7==0) and (i%5!=0):
        l.append(str(i))
    print (','.join(l))
```

```
2002,2009,2016,2023,2037,2044,2051,2058,2072,2079,2086,2093,2107,2114,2121,2128,2142,2149,21
```

b. **Which will have a list of values and then input a number to check if the value exist in the list or not.**

Code + Output:

```
]  # Q3(b)List exist or not
   l=[ 1,2,3,4,5,6 ]
   for k in range(2):
     i=int(input("Enter number to check in the list: "))
     if i in l:
       print("Number Exists in list")
     else:
       print("Number Not exists in list")
```

```
Enter number to check in the list: 5
Number Exists in list
Enter number to check in the list: 79
Number Not exists in list
```

Question 4

Write a program to:

a. **Input 5 numbers in the list and print in ascending order.**

Code + Output:

```
[ ]  # Q4(a) input 5 no.s and Sort
     numbers = []
     print("Enter 5 numbers: ")
     for i in range(5):
       n = int(input())
       numbers.append(n)
       numbers.sort()
     print("Numbers in ascending order:")
     for i in numbers:
       print(i,end=" ")
```

```
Enter 5 numbers:
8
4
6
3
1
Numbers in ascending order:
1 3 4 6 8
```

b. **Insert a value in the list at particular position.**

Code + Output:

```
]  # Q4(b) Insert value at particular pos
   l=[1,2,3,"Hitesh",True,69.69]
   val=input("Enter value you want to insert: ")
   index=int(input("Enter index where you want to insert: "))
   l.insert(index,val)
   print(l)
```

```
Enter value you want to insert: Walia
Enter index where you want to insert: 4
[1, 2, 3, 'Hitesh', 'Walia', True, 69.69]
```

c. Count elements in the list until the occurrence of the first tuple. In the list.

Code + Output:

```
# Q4(c) Count elements untill tuple appears
l=[1,2,3,4,(6,7,8,),"klkl",False]
count=0
for i in l:
  if(isinstance(i,tuple)):
    break
  count=count+1
print("No. of elements untill tuple appears:",count)
```

```
No. of elements untill tuple appears: 4
```

Question 5

Write a program to:

a. **Replace an empty tuple with another tuple or list.**

Code + Output:

```
[ ]  # Q5(a)replace empty tuple with list/tuple

     def replacee(l, rep):
       for i in range(len(l)):
         if isinstance(l[i], tuple) and len(l[i]) == 0:
             l[i] = rep
             break
       return l

     my_list = [1, 2, (), 4, 5]
     print("Orginal list: ",my_list)
     rep = ('h', 'l', 'c','i')
     res = replacee(my_list, rep)
     print("Updated list:", res)
```

```
Orginal list:  [1, 2, (), 4, 5]
Updated list: [1, 2, ('h', 'l', 'c', 'i'), 4, 5]
```

b. **Replace last element of a tuple.**

Code + Output:

```
[ ]  # Q5(b) Replace last element of a tuple.

     def rep_last_el(tuple_data, new_element):
       list_data = list(tuple_data)
       list_data[-1] = new_element
       new_tuple = tuple(list_data)
       return new_tuple

     t = (1, 2, 3, 4, 5)
     print("Original tuple:",t)
     res = rep_last_el(t, 6)
     print("Updated tuple:",res)
```

```
Original tuple: (1, 2, 3, 4, 5)
Updated tuple: (1, 2, 3, 4, 6)
```

Question 6

Write a program to:

a. **implement Linear search on list and tuple.**

```
[ ]  # Q6(a)(i) Linear Search on Lists
     n=int(input("Enter number of elements of list: "))
     l=[]
     print("Enter elements: ")
     for i in range(n):
         item=int(input())
         l.append(item)

     print("List is: ",l)
     for i in range(2):
       flag=1
       el=int(input("Enter element to search: "))
       for i in range(len(l)):
           if(l[i]==el):
             print("Element found at index",i)
             flag=0
             break;
       if flag==1:
         print("Element not found :(")
```

```
Enter number of elements of list: 5
Enter elements:
1
3
2
5
4
List is:  [1, 3, 2, 5, 4]
Enter element to search: 5
Element found at index 3
Enter element to search: 34
Element not found :(
```

```
[ ]  # Q6(a)(ii). Linear Search on Tuple
     tup=(4,2,3,1,7,8)
     for i in range(2):
       el=int(input("Enter element to search: "))
       if el in tup:
         print("Element found :)")
       else:
         print("Element not found :(")
```

```
Enter element to search: 3
Element found :)
Enter element to search: 99
Element not found :(
```

b. print Fibonacci series.

```
[ ]  # Q6(b). Write a program to print fibonacci series up to a given number
     a=c=0
     b=1
     print("Enter n upto which you want Fibonacci Series: ",end="")
     n=int(input())
     print(a,end=" ")
     print(b,end=" ")
     for i in range(2,n):
         c=a+b
         print(c,end=" ")
         a=b
         b=c
```

```
Enter n upto which you want Fibonacci Series: 8
0 1 1 2 3 5 8 13
```

Question 7

Wap to create a function that accepts abbreviations of states as args and returns the full name of the states, return null if no mapping is found.

```
[ ] # Q7 abb and states
    def return_state_name(abb):
        dict={"DL":"Delhi", "UP":"Uttar Pradesh", "HP":"Himachal Pradesh" ,"GJ":"Gujrat",
              "PB":"Punjab", "JK":"Jammu and Kashmir","RJ":"Rajashtan", "AS":"Assam"}
        return dict[abb]
    abb=input("Enter abbreviation of state: ")
    print(return_state_name(abb))
```

```
Enter abbreviation of state: UP
Uttar Pradesh
```

Question 8

Write a function:

a. **To print all powers of 2 in the range 1-12 (inclusive).i.e including 12.**

```
[ ] # Q8(a). Write a function to print all the powers of two, up to and including the twelfth power.

    def func2():
      for i in range(1,13):
        print("2 raise to power",i,"is:",2**i)
    func2()
```

```
2 raise to power 1 is: 2
2 raise to power 2 is: 4
2 raise to power 3 is: 8
2 raise to power 4 is: 16
2 raise to power 5 is: 32
2 raise to power 6 is: 64
2 raise to power 7 is: 128
2 raise to power 8 is: 256
2 raise to power 9 is: 512
2 raise to power 10 is: 1024
2 raise to power 11 is: 2048
2 raise to power 12 is: 4096
```

b. **That accepts lowercase words and returns uppercase.**

```
[ ] # Q8(b). Define a function that accepts lowercase words and returns uppercase words.
    def func3():
      s=input("Enter any string: ")
      if(s.islower()):
        s=s.upper()
        print(s)
      else:
        print("Enter string only in lower case...")
        func3()

    func3()
```

```
Enter any string: Hitesh
Enter string only in lower case...
Enter any string: hitesh
HITESH
```

Question 9

Write a program to create a function show_employee() using the conditions:

i. It should accept the employee's name and salary and display both.

ii. If the salary is missing in the function call then assign default value 50000 to salary

```
[ ] # Q9.Write a program to create a function show_employee() using the  conditions:
    #     (a)It should accept the employee's name and salary and display both.
    #     (b)If the salary is missing in the function call then assign default value 50000 to salary

    def show_employee(name,sal=50000):
      print("Employee's Name:",name)
      print("Employee's Salary:",sal)

    show_employee("Hitesh",69000)
    show_employee("Raghav",59000)
    show_employee("Aman")
```

```
Employee's Name: Hitesh
Employee's Salary: 69000
Employee's Name: Raghav
Employee's Salary: 59000
Employee's Name: Aman
Employee's Salary: 50000
```

Question 10

Write a program:

a. That accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.

```
[ ] # Q10(a)# Write a Python program that accepts a hyphen-separated sequence of words as input and
    #  prints the words in a hyphen-separated sequence after sorting them alphabetically.

    words = input("Enter a hyphen-separated sequence of words: ")
    word_list = words.split("-")
    print(word_list)
    sorted_words = sorted(word_list)
    print(sorted_words)
    sorted_sequence = "-".join(sorted_words)
    print("Sorted words:", sorted_sequence)
```

```
Enter a hyphen-separated sequence of words: def-klmnop-abc-z
['def', 'klmnop', 'abc', 'z']
['abc', 'def', 'klmnop', 'z']
Sorted words: abc-def-klmnop-z
```

b. To add some days to your present date and print the date added.

```
[ ] # Q10(b) Write a python program to add some days to your present date and print the date added.
    from datetime import datetime, timedelta
    def add_days_to_date(num_days):
     current_date = datetime.now().date()
     print("Current Date: ",current_date)
     new_date = current_date + timedelta(days=num_days)
     return new_date

    num_days = int(input("Enter the number of days to add: "))
    result_date = add_days_to_date(num_days)
    print("New date:", result_date)
```

```
Enter the number of days to add: 3
Current Date:  2023-05-23
New date: 2023-05-26
```

Question 11

Write a Python function that takes two lists and returns True if they are equal otherwise false

```
[ ] # Q11. Write a Python function that takes two lists and returns True if they are equal otherwise false
    def func9(l1,l2):
      if(l1==l2):
        return True
      else:
        return False

    l1=[10,15,17,20]
    l2=[10,15,17,20]
    res=func9(l1,l2)
    print(res)

    l3=[11,15,17,20]
    l4=[10,15,17,20]
    res=func9(l3,l4)
    print(res)
```

```
True
False
```

Question 12

Wap to print the following patterns:

a.

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

```
[ ]  # Q12(a)
     n=int(input("Enter number of rows: "))
     m=int(input("Enter number of columns: "))
     for i in range(n):
       for j in range(m):
         print("* ",end="")
       print()
```

```
Enter number of rows: 5
Enter number of columns: 5
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

b.

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

4 4 4 4 4

5 5 5 5 5

```
[ ]  # Ques12(b)
     n=int(input("Enter number of rows: "))
     m=int(input("Enter number of columns: "))
     for i in range(1,n+1):
       for j in range(m):
         print(i,end="")
       print()
```

```
Enter number of rows: 5
Enter number of columns: 5
11111
22222
33333
44444
55555
```

c.

1 2 3 4 5 6 7

1 2 3 4 5 6 7

1 2 3 4 5 6 7

1 2 3 4 5 6 7

1 2 3 4 5 6 7

1 2 3 4 5 6 7

1 2 3 4 5 6 7

```
[ ]  # Ques12(c)
     n=int(input("Enter number of rows: "))
     m=int(input("Enter number of columns: "))
     for i in range(1,n+1):
       for j in range(1,m+1):
         print(j,end="")
       print()
```

```
Enter number of rows: 7
Enter number of columns: 7
1234567
1234567
1234567
1234567
1234567
1234567
1234567
```

d.

A A A A A A A

B B B B B B B

C C C C C C C

D D D D D D D

E E E E E E E

F F F F F F F

G G G G G G G

```
[ ] # Ques12(d)
    n=int(input("Enter number of rows: "))
    m=int(input("Enter number of columns: "))
    for i in range(n):
      for j in range(m):
        print(chr(i+65),end="")
      print()
```

```
Enter number of rows: 7
Enter number of columns: 7
AAAAAAA
BBBBBBB
CCCCCCC
DDDDDDD
EEEEEEE
FFFFFFF
GGGGGGG
```

e.

```
 *
* *
* * *
* * * *
* * * * *
```

```
[ ]  # Q12(e)
     n=5
     # n=int(input("Enter number of rows: "))
     for i in range(n):
       for k in range(i,n+1):
           print("",end="")
       for j in range(i+1):
         print("*",end="")
       print()
```

```
*
**
***
****
*****
```

f.

```
    *
   * *
  * * *
 * * * *
* * * * *
```

```
[ ] # Q12(f)
    n=5
    for i in range(n):
      for k in range(i,n+1):
          print(" ",end="")
      for j in range(i+1):
        print("*",end="")
      print()
```

```
    *
   **
  ***
 ****
*****
```

g.

```
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

```
[2] # Q12(g)
    n=5
    for i in range(n):
      for k in range(i,n+1):
          print(" ",end="")
      for j in range(2*i+1):
        print("* ",end="")
      print()
```

```
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

i.

1 1 1 1 1

2 2 2 2

3 3 3

4 4

5

```
# Q12(i)
num = 1
for i in range(5, 0, -1):
 print(" " * (5 - i) + (str(num) + " ") * i)
 num += 1
```

```
1 1 1 1 1
 2 2 2 2
  3 3 3
   4 4
    5
```

Question 13

Write a Python program to reverse a string using function .

Sample String : "1234abcd"

Expected Output : "dcba4321

```
[ ] # Q13. Write a Python program to reverse a string using function .
    def rev(s):
      return s[::-1]
    s="1234abcd"
    print("Original string:",s)
    s=rev(s)
    print("Reversed string:",s)
```

```
Original string: 1234abcd
Reversed string: dcba4321
```

Question 14

Program to:

a. compute gcd of two numbers recursively in Python.

```
[ ]  # Q14(a)compute gcd of two numbers recursively in Python.
     def gcd(a,b):
       if(b==0):
         return a
       return gcd(b,a%b)

     n1=int(input("Enter first number: "))
     n2=int(input("Enter second number: "))
     print("GCD of",n1,"and",n2,"is:",gcd(n1,n2))
```

```
Enter first number: 36
Enter second number: 24
GCD of 36 and 24 is: 12
```

b. to find factorial of a number using Recursion.

```
[ ]  # Q14(b) to find factorial of a number using Recursion.
     def fact(n):
       if(n==1 or n==0): return n
       return n * fact(n-1)

     n=int(input("Enter any number: "))
     print("Factorial is:",fact(n))
```

```
Enter any number: 7
Factorial is: 5040
```

Question 15

Python program to:

a. convert decimal into other number systems

```
[ ]  # Q15(a). convert decimal into other number systems
     decimal_num = int (input("Enter a decimal number: "))
     binary_num = bin(decimal_num)
     print("Binary:", binary_num)
     octal_num = oct(decimal_num)
     print("Octal:", octal_num)
     hex_num = hex(decimal_num)
     print("Hexadecimal:", hex_num)
```

```
Enter a decimal number: 69
Binary: 0b1000101
Octal: 0o105
Hexadecimal: 0x45
```

b. Make a Simple Calculator using functions.

```
[ ]  # Q15(b). Make a Simple Calculator using functions.
     def add(a,b):
       return a + b
     def subtract(a,b):
       return a - b
     def multiply(a,b):
       return a * b
     def divide(a,b):
       return a / b
     print("******** Welcome to the Calculator ********")
     while True:
       print()
       print("Enter your choice: ")
       print("1.Add")
       print("2.Subtract")
       print("3.Multiply")
       print("4.Divide")
       print("5.Exit")
       choice = input("Enter your choice: ")
       if choice == '5':
         print("Program Terminated!!!")
         break
       num1 = float(input("Enter first number: "))
       num2 = float(input("Enter second number: "))
```

```
[ ]   if choice == '1':
        result = add(num1, num2)
        print("Addition is:", result)
      elif choice == '2':
        result = subtract(num1, num2)
        print("Subtraction is:", result)
      elif choice == '3':
        result = multiply(num1, num2)
        print("Multiplication is:", result)
      elif choice == '4':
        if num2 != 0:
          result = divide(num1, num2)
          print("Division is:", result)
        else:
            print("Error: Cannot divide by zero!")
      else:
          print("Invalid choice!!!")
```

```
[ ]  ******** Welcome to the Calculator ********

     Enter your choice:
     1.Add
     2.Subtract
     3.Multiply
     4.Divide
     5.Exit
     Enter your choice: 1
     Enter first number: 2
     Enter second number: 3
     Addition is: 5.0

     Enter your choice:
     1.Add
     2.Subtract
     3.Multiply
     4.Divide
     5.Exit
     Enter your choice: 2
     Enter first number: 6
     Enter second number: 3
     Subtraction is: 3.0
```

```
Enter your choice:
1.Add
2.Subtract
3.Multiply
4.Divide
5.Exit
Enter your choice: 3
Enter first number: 7
Enter second number: 9
Multiplication is: 63.0

Enter your choice:
1.Add
2.Subtract
3.Multiply
4.Divide
5.Exit
Enter your choice: 4
Enter first number: 98
Enter second number: 6
Division is: 16.333333333333332
```

Question 16

Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).

```
[ ]  # Q16 Right Triangle or not (Pythagorean Theorem).
     base=float(input("Enter base: "))
     perp=float(input("Enter perpendicular: "))
     hypo=float(input("Enter hypotenuse: "))
     print(f"Base is : {base}")
     print(f"Perpendicular is : {perp}")
     print(f"Hypotenuse is : {hypo}")

     if hypo**2==(base**2 + perp**2):
       print("It is a right angled triangle :)")
     else:
       print("It is not a right angled triangle :(")
```

```
Enter base: 5
Enter perpendicular: 12
Enter hypotenuse: 13
Base is : 5.0
Perpendicular is : 12.0
Hypotenuse is : 13.0
It is a right angled triangle :)
```

Question 17

Write a python program to define a module to find Fibonacci Numbers and import the module to another program.

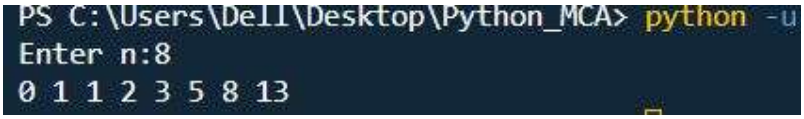
fib_module.py

```
def fibo(n):
    if(n==0 or n==1):
        return n
    return fibo(n-1) + fibo(n-2)
```

fib.py

```
import fib_module
n=int(input("Enter n:"))
for i in range(n):
    print(fib_module.fibo(i),end=" ")
```

Output



Question 18

Write a program:

a. using decorator to perform division.

```
[1]  # Q18(a) Decorator
     def div(a,b):
       print(a/b)

     def deco_func(func):
       def inner(a,b):
         if a<b:
           a,b=b,a
         return func(a,b)
       return inner

     calling_func = deco_func(div)
     num=float(input("Enter numerator: "))
     den=float(input("Enter denominator: "))
     calling_func(num,den)
```

```
Enter numerator: 24
Enter denominator: 4
6.0
```

b. Iterator

```
# Q18(b) Iterator
def div(a,b):
  print(a/b)

def deco_func(func):
  def inner(a,b):
    if a<b:
      a,b=b,a
    return func(a,b)
  return inner

calling_func = deco_func(div)
user_inputs=iter(input("Enter numerator and denominator: ").split())
num=float(next(user_inputs))
den=float(next(user_inputs))
calling_func(num,den)
```

```
Enter numerator and denominator: 34 17
2.0
```

c. Generator.

```
[8]  # 18(c) Generator
     def div(a,b):
       yield a/b

     num=float(input("Enter numerator: "))
     den=float(input("Enter denominator: "))
     print(div(num,den))
     res=div(num,den)
     print(next(res))
```

```
Enter numerator: 5
Enter denominator: 2
<generator object div at 0x7f2ccdb03ae0>
2.5
```

Question 19

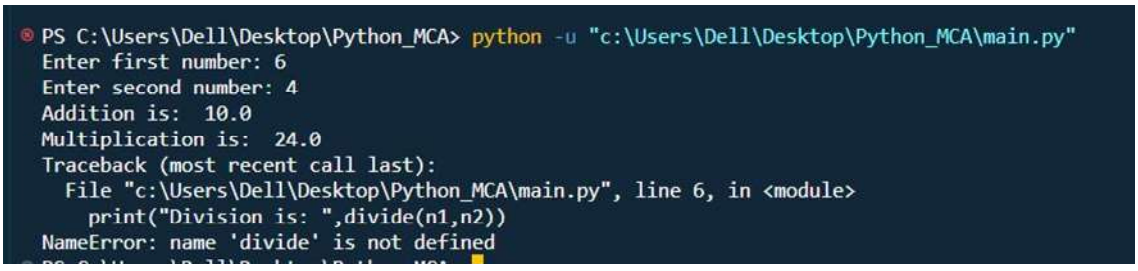
Write a python program to define a module and import a specific function in that module to another program.

operations.py (module)

```
def add(a,b):
    return a+b

def subtract(a,b):
    return a-b

def divide(a,b):
    return a/b

def multiply(a,b):
    return a*b
```

main.py

```
from operations import add,multiply
n1=float(input("Enter first number: "))
n2=float(input("Enter second number: "))
print("Addition is: ",add(n1,n2))
print("Multiplication is: ",multiply(n1,n2))
print("Division is: ",divide(n1,n2))
```

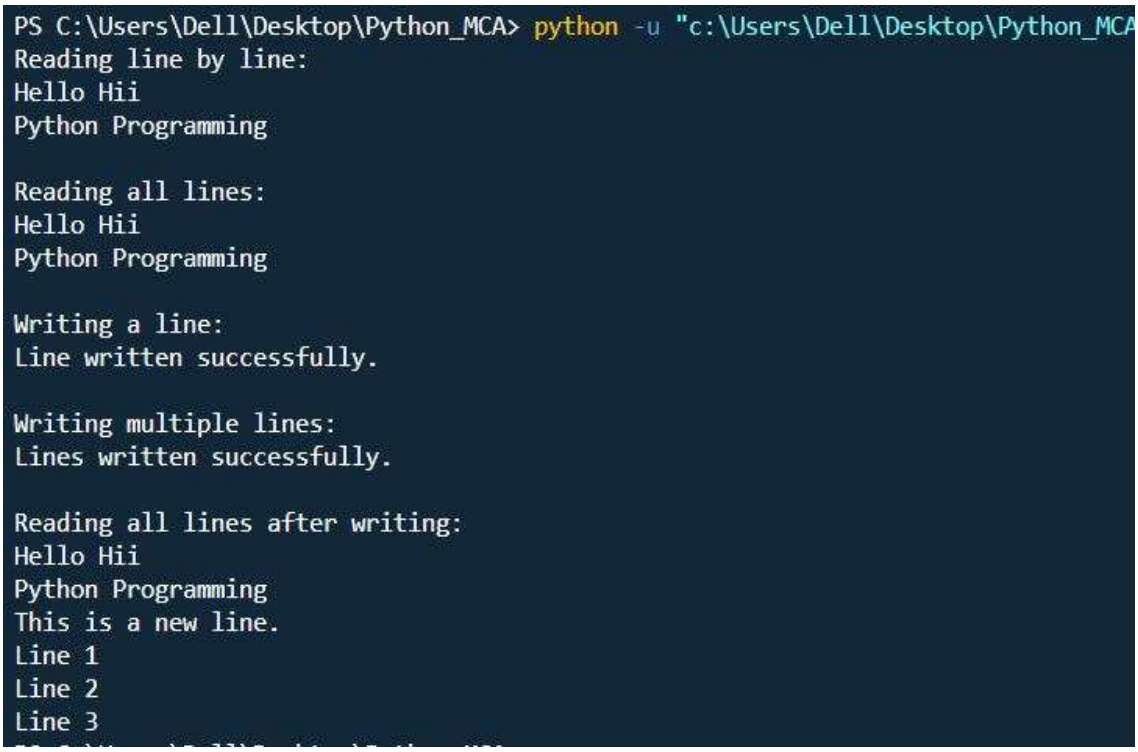


Question 20

Write a program in Python to implement readline, readlines, writeline and writelines using file handling mechanisms.

```
def read_line_by_line(file_name):
    with open(file_name, 'r') as file:
        line = file.readline()
        while line:
            print(line.strip())
            line = file.readline()


def read_all_lines(file_name):
    with open(file_name, 'r') as file:
        lines = file.readlines()
        for line in lines:
            print(line.strip())


def write_line(file_name, text):
    with open(file_name, 'a') as file:
        file.write(text + '\n')


def write_lines(file_name, lines):
    with open(file_name, 'a') as file:
        file.writelines(lines)


file_name = 'example.txt'


print("Reading line by line:")
read_line_by_line(file_name)
print()
```

```
print("Reading all lines:")
read_all_lines(file_name)
print()

print("Writing a line:")
write_line(file_name, "This is a new line.")
print("Line written successfully.")
print()

print("Writing multiple lines:")
lines_to_write = [
    "Line 1\n",
    "Line 2\n",
    "Line 3\n"
]
write_lines(file_name, lines_to_write)
print("Lines written successfully.")
print()

print("Reading all lines after writing:")
read_all_lines(file_name)
```


PS C:\Users\Dell\Desktop\Python_MCA> python -u "c:\Users\Dell\Desktop\Python_MCA
Reading line by line:
Hello Hii
Python Programming
Reading all lines:
Hello Hii
Python Programming
Writing a line:
Line written successfully.
Writing multiple lines:
Lines written successfully.
Reading all lines after writing:
Hello Hii
Python Programming
This is a new line.
Line 1
Line 2
Line 3

Question 21

a) Write a Python class to reverse a string word by word.

```
[23] # Q21(a)
     class A:
       def reverse(self, string):
         words = string.split()
         reversed_words = words[::-1]
         reversed_string = ' '.join(reversed_words)
         return reversed_string

     reverser = A()
     input_string = "Good Morning Mango"
     rev = reverser.reverse(input_string)
     print(rev)
```

```
Mango Morning Good
```

b) Write a program to show multiple inheritance in python.

```
#21. b) write a program to show multiple inheritance in python.
class Person:
  def __init__(self, name):
    self.name = name
  def display_name(self):
    print(f"Name: {self.name}")

class Course:
  def __init__(self, course_name):
    self.course_name = course_name
  def display_course(self):
    print(f"Course: {self.course_name}")

class Student(Person, Course):
  def __init__(self, name, course_name, student_id):
    Person.__init__(self, name)
    Course.__init__(self, course_name)
    self.student_id = student_id
  def display_info(self):
    self.display_name()
    self.display_course()
    print(f"Student ID: {self.student_id}")

st = Student("ABCD", "Python", "254")
st.display_info()
```

```
Name: ABCD
Course: Python
Student ID: 254
```

Question 22

Demonstrate the following functions/methods which operates on dictionary in Python with suitable examples:

i) dict( ) ii) len( ) iii) clear( ) iv) get( )

v) pop( ) vi)popitem( ) vii) keys( ) viii) values()

ix) items( ) x) copy( ) xi) update( )

dict(): This function is used to create new dictionaries.

```
empty_dict = dict()
print(empty_dict)

student = dict(name='John', age=22, grade='A')
print(student)
```

```
{}
{'name': 'John', 'age': 22, 'grade': 'A'}
```

Len(): This function returns the key-value pairs in a dictionary.

```
student = {'name': 'John', 'age': 20, 'grade': 'A'}
print(len(student))
```

```
3
```

clear(): This method removes all key-value pairs from a dictionary, making it empty.

```
student = {'name': 'John', 'age': 20, 'grade': 'A'}
student.clear()
print(student)
```

```
{}
```

get() - This method returns the value associated with a given key in a dictionary. If the key is not found, it returns a default value (optional argument).

```
student = {'name': 'John', 'age': 20, 'grade': 'A'}
print(student.get('name'))
print(student.get('address'))
print(student.get('address', 'N/A'))
```

```
John
None
N/A
```

pop() - This method removes the key-value pair with the specified key from a dictionary and returns the corresponding value.

```
student = {'name': 'Markie De-malieo', 'age': 20, 'grade': 'A'}
age = student.pop('age')
print(age)
print(student)
```

```
20
{'name': 'Markie De-malieo', 'grade': 'A'}
```

popitem() - This method removes and returns an arbitrary key-value pair from a dictionary.

```
student = {'name': 'Karl', 'age': 21, 'grade': 'B'}
key, value = student.popitem()
print(key, value)
print(student)
```

```
grade B
{'name': 'Karl', 'age': 21}
```

keys() - This method returns a list of all the keys in a dictionary.

```
student = {'name': 'Pearl', 'age': 19, 'grade': 'A-'}
keys = student.keys()
print(keys)
```

```
dict_keys(['name', 'age', 'grade'])
```

values() - This method returns a list of all the values in a dictionary.

```
student = {'name': 'RRS', 'age': 20, 'grade': 'A'}
values = student.values()
print(values)
```

```
dict_values(['RRS', 20, 'A'])
```

items() - This method returns a list of tuples containing all the key-value pairs in a dictionary.

```
student = {'name': 'HW', 'age': 21, 'grade': 'A'}
items = student.items()
print(items)
```

```
dict_items([('name', 'John'), ('age', 20), ('grade', 'A')])
```

copy() - This method creates a shallow copy of a dictionary.

```
student = {'name': 'BR', 'age': 21, 'grade': 'A'}
student_copy = student.copy()
print(student_copy)
```

```
{'name': 'John', 'age': 20, 'grade': 'A'}
```

Question 23

Demonstrate the following functions/methods which operates on sets in Python with suitable examples:

i) add( ) ii) update( ) iii) copy( ) iv) pop( )

v) remove( ) vi)discard( ) vii) clear( ) viii) union()

ix) intersection( ) x) difference( )

add() - This method is used to add an element to a set

```
#1
fruits = {'apple', 'banana', 'cherry','watermelon'}
fruits.add('orange')
print(fruits)
```

```
{'cherry', 'watermelon', 'banana', 'orange', 'apple'}
```

update() - This method is used to add multiple elements to a set.

```
#2
fruits = {'apple', 'banana', 'cherry'}
fruits.update(['orange', 'mango'])
print(fruits)
```

```
{'cherry', 'banana', 'orange', 'apple', 'mango'}
```

copy() - This method creates a shallow copy of a set.

```
#3
fruits = {'apple', 'banana', 'cherry','kiwi'}
fruits_copy = fruits.copy()
print(fruits_copy)
```

```
{'cherry', 'banana', 'kiwi', 'apple'}
```

pop() - This method removes and returns an arbitrary element from a set.

```
#4
fruits = {'apple', 'banana', 'cherry'}
removed_fruit = fruits.pop()
print(removed_fruit)
print(fruits)
```

```
cherry
{'banana', 'apple'}
```

remove() - This method removes a specific element from a set. Raises a KeyError if the element is not found**.**

```
#5
fruits = {'apple', 'banana', 'cherry'}
fruits.remove('banana')
print(fruits)
```

```
{'cherry', 'apple'}
```

discard() - This method removes a specific element from a set, if it is present. Does not raise an error if the element is not found.

```
fruits = {'apple', 'banana', 'cherry'}
fruits.discard('banana')
print(fruits)
```

```
{'cherry', 'apple'}
```

clear() - This method removes all elements from a set, making it empty.

```
#7
fruits = {'apple', 'banana', 'cherry'}
fruits.clear()
print(fruits)
```

```
set()
```

union() - This method returns a new set that contains all unique elements from two or more sets.

```
#8
set1 = {1, 2, 3}
set2 = {3, 4, 5}
union_set = set1.union(set2)
print(union_set)
```

```
{1, 2, 3, 4, 5}
```

intersection() - This method returns a new set that contains common elements between two or more sets.

```
set1 = {1,2,3,4,5}
set2 = {3,4,5,7,6,9}
intersection_set = set1.intersection(set2)
print(intersection_set)
```

```
{3, 4, 5}
```

difference() - This method returns a new set that contains elements present in the first set but not in the other sets.

```
set1 = {1, 2, 3, 4}
set2 = {3, 4, 5,6 ,7}
difference_set = set1.difference(set2)
print(difference_set)
```

```
{1, 2}
```

Question 24

Demonstrate lambda functions in Python with suitable example programs.

```
[22] # Q24.
      # Addition
     sum = lambda x, y: x + y
     result = sum(10, 30)
     print(f"Addition of 10,30 is : {result}")

     # square of a number
     square = lambda x: x**2
     result = square(7)
     print(f"Square of 7 is : {result}")
```

```
Addition of 10,30 is : 40
Square of 7 is : 49
```

Question 25

Explain following commands in pandas libaray:

Query() , head(), shape() , Describe() , Tail(), iloc(0, loc(), drop(), min(0 , Max(), Groupby()

```
[19] import pandas as pd
     # Creating a sample DataFrame
     data = {'Name': ['OPU', 'JKL', 'PQR', 'ABC', 'XYZ'],
      'Age': [25, 30, 35, 28, 32],
       'City': ['New York', 'London', 'Paris', 'London', 'Paris'],
      'Salary': [50000, 60000, 70000, 55000, 65000]}
     df = pd.DataFrame(data)
     print("---Querying the DataFrame---")
     filtered_df = df.query("Age >= 30")
     print(filtered_df)
     print()
     print("---Retrieving the first few rows---")
     head_df = df.head(3)
     print(head_df)
     print()
     print("---Getting the shape of the DataFrame---")
     df_shape = df.shape
     print(df_shape)
     print()
     print("---Descriptive statistics of the DataFrame---")
     df_description = df.describe()
     print(df_description)
     print()
     print("---Retrieving the last few rows---")
     tail_df = df.tail(2)
```

```
print(tail_df)
print()
print("---Accessing specific elements using iloc---")
element = df.iloc[1, 2]
print(element)
print()
print("---Accessing specific elements using loc---")
element = df.loc[3, 'Name']
print(element)
print()
print("---Dropping a column from the DataFrame---")
dropped_df = df.drop('Salary', axis=1)
print(dropped_df)
print()
print("---Calculating the minimum Age---")
min_value = df['Age'].min()
print(min_value)
print()
print("---Calculating the maximum Salary---")
max_value = df['Salary'].max()
print(max_value)
print()

print("---Grouping the data by 'City'---")
grouped_data = df.groupby('City')
for name, city in grouped_data:
    print(name)
    print(city)
```

```
---Querying the DataFrame---
  Name  Age    City  Salary
1  JKL   30  London   60000
2  PQR   35   Paris   70000
4  XYZ   32   Paris   65000

---Retrieving the first few rows---
  Name  Age      City  Salary
0  OPU   25  New York   50000
1  JKL   30    London   60000
2  PQR   35     Paris   70000

---Getting the shape of the DataFrame---
(5, 4)

---Descriptive statistics of the DataFrame---
             Age       Salary
count   5.000000      5.00000
mean   30.000000  60000.00000
std     3.807887   7905.69415
min    25.000000  50000.00000
25%    28.000000  55000.00000
50%    30.000000  60000.00000
75%    32.000000  65000.00000
max    35.000000  70000.00000

---Retrieving the last few rows---
  Name  Age    City  Salary
3  ABC   28  London   55000
4  XYZ   32   Paris   65000
```

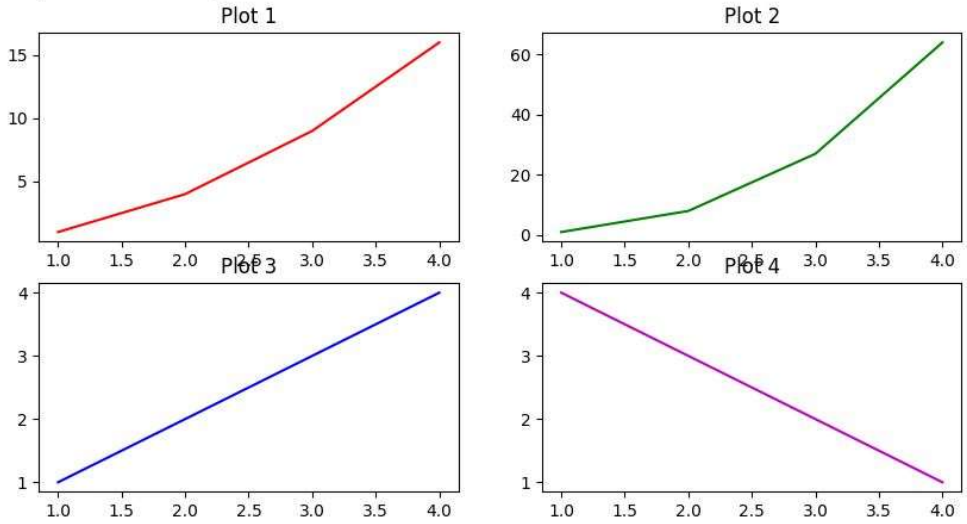
```
---Accessing specific elements using iloc---
London

---Accessing specific elements using loc---
ABC

---Dropping a column from the DataFrame---
  Name  Age      City
0  OPU   25  New York
1  JKL   30    London
2  PQR   35     Paris
3  ABC   28    London
4  XYZ   32     Paris

---Calculating the minimum Age---
25

---Calculating the maximum Salary---
70000

---Grouping the data by 'City'---
London
  Name  Age    City  Salary
1  JKL   30  London   60000
3  ABC   28  London   55000
New York
  Name  Age      City  Salary
0  OPU   25  New York   50000
Paris
  Name  Age   City  Salary
2  PQR   35  Paris   70000
4  XYZ   32  Paris   65000
```

Question 26

a) Write a program to demonstrate subplots and multiple plots in matplotlib.

```
[11] # 26. a) Write a program to demonstrate subplots and multiple plots in
     # matplotlib.
     # b) Demonstrate pie chart and scatter plot in matplotlib.

     import matplotlib.pyplot as plt
     %matplotlib inline
     #(a) Subplots and Multiple Plots
     fig, axes = plt.subplots(2, 2, figsize=(10, 5))
     axes[0, 0].plot([1, 2, 3, 4], [1, 4, 9, 16], 'r')
     axes[0, 0].set_title('Plot 1')

     axes[0, 1].plot([1, 2, 3, 4], [1, 8, 27, 64], 'g')
     axes[0, 1].set_title('Plot 2')

     axes[1, 0].plot([1, 2, 3, 4], [1, 2, 3, 4], 'b')
     axes[1, 0].set_title('Plot 3')

     axes[1, 1].plot([1, 2, 3, 4], [4, 3, 2, 1], 'm')
     axes[1, 1].set_title('Plot 4')
```

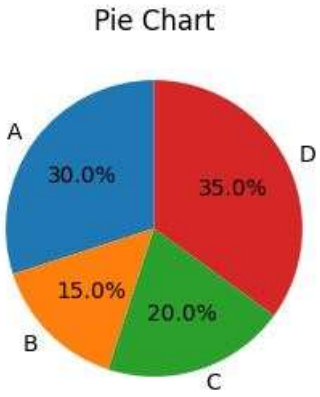
```
[11] Text(0.5, 1.0, 'Plot 4')
```

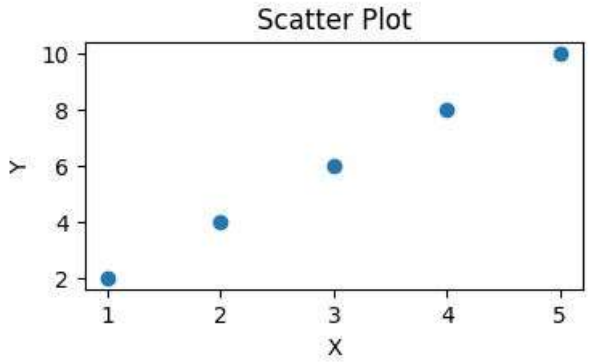


b) Demonstrate pie chart and scatter plot in matplotlib

```
[18] #26(b) Pie chart
     sizes = [30, 15, 20, 35]
     labels = ['A', 'B', 'C', 'D']
     plt.figure(figsize=(3, 3))
     plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
     plt.title('Pie Chart')
     plt.show()

     # Scatter plot
     x = [1, 2, 3, 4, 5]
     y = [2, 4, 6, 8, 10]
     plt.figure(figsize=(4, 2))
     plt.scatter(x, y)
     plt.title('Scatter Plot')
     plt.xlabel('X')
     plt.ylabel('Y')
     plt.show()
```

[18]





Question 27

Write a program to show joining in Numpy arrays , intersection and difference in Numpy.

```
[21] import numpy as np
     arr1 = np.array([10, 20, 30])
     arr2 = np.array([7, 5, 90])
     arr_join = np.concatenate((arr1, arr2))
     print("Joined Array:")
     print(arr_join)

     arr3 = np.array([ 3, 4, 5])
     arr4 = np.array([4, 50, 67, 3, 8])
     arr_intersect = np.intersect1d(arr3, arr4)
     print("\nIntersection:")
     print(arr_intersect)

     arr5 = np.array([1, 2, 3, 4, 5])
     arr6 = np.array([4, 5, 6, 7, 8])
     arr_diff = np.setdiff1d(arr5, arr6)
     print("\nDifference:")
     print(arr_diff)
```

```
Joined Array:
[10 20 30  7  5 90]

Intersection:
[3 4]

Difference:
[1 2 3]
```