Slip No -1

Q1) Create a class circle having members radius. Use operator overloading to add the radius of two circle objects. Also display the area of circle. [15 marks]

program:

```
import math

class Circle:

    def __init__(self, radius):
        self.__radius = radius

    def setRadius(self, radius):
        self.__radius = radius

    def getRadius(self):
        return self.__radius

    def area(self):
        return math.pi * self.__radius ** 2

    def __add__(self, another_circle):
        return Circle( self.__radius + another_circle.__radius )

c1 = Circle(4)
print(c1.getRadius())

c2 = Circle(5)
print(c2.getRadius())
c3 = c1 + c2 # This became possible because we have overloaded + operator by adding a method named __add__
print(c3.getRadius())
```

output:**********

4
5
9

Q2) **Write a Python program to accept n numbers in list and remove duplicates from a list.**

program:

```
dupList = []

listNumber = int(input("Enter the Total List Items = "))
for i in range(1, listNumber + 1):
    listValue = int(input("Enter the %d List Item = " %i))
    dupList.append(listValue)

print("List Items = ", dupList)

uniqList = []

for val in dupList:
    if val not in uniqList:
        uniqList.append(val)

print("List Items after removing Duplicates = ", uniqList)
```

**************************output:**************************

```
Enter the Total List Items = 5
Enter the 1 List Item = 12
Enter the 2 List Item = 23
Enter the 3 List Item = 12
Enter the 4 List Item = 43
Enter the 5 List Item = 34
List Items =  [12, 23, 12, 43, 34]
List Items after removing Duplicates =  [12, 23, 43, 34]
```

Slip No -2

Q1 Write a python program to convert tuple of string values to tuple of integer values. Original tuple values : ((‘333’,’33’),(‘1234’,’55’)) New tuple value (333,33), (1234,55))
[15 marks]

program-

```
def tuple_int_str(tuple_str):
    result = tuple((int(x[0]), int(x[1])) for x in tuple_str)
    return result

tuple_str =  (('333', '33'), ('1416', '55'))
print("Original tuple values:")
print(tuple_str)
print("\nNew tuple values:")
print(tuple_int_str(tuple_str))
```

**********output**********

```
Original tuple values:
(('333', '33'), ('1416', '55'))

New tuple values:
((333, 33), (1416, 55))
```

Q2) Define class Rectangle which can be constructed by length and width. The Rectangle class has method which can compute the area and perimeter

program:

```
class Rectangle:
    def __init__(self, length, breadth):
        self.length = length
        self.breadth = breadth

    def display(self):
        print ("Length of Rectangle is: ", self.length)
        print ("Breadth of Rectangle is: ", self.breadth)

    def area(self):
        return (self.length*self.breadth)

    def perimeter(self):
        return (2*self.length + 2*self.breadth)
```

```
l = int (input("Enter the length of the Rectangle: "))
b = int (input("Enter the breadth of rectangle: "))

r1 = Rectangle(l , b)
print ("Rectangle object details are:")
r1.display()
print("")
print ("Area of Rectangle is: ", r1.area())
print("")
print ("The Perimeter of the Rectangle is: ", r1.perimeter())
```

**********output**********

Enter the length of the Rectangle: 5

```
Enter the breadth of rectangle: 4
Rectangle object details are:
Length of Rectangle is:  5
Breadth of Rectangle is:  4

Area of Rectangle is:  20

The Perimeter of the Rectangle is:  18
```

Slip no-3

Q1 Define class circle which has method to can compute the area and perimeter(use parameterized constructor).

Program:

```
 class Circle():
    def __init__(self, r):
        self.radius = r
# Program to multiply two matrices using nested loops

# 3x3 matrix
X = [[12,7,3],
    [4 ,5,6],
    [7 ,8,9]]
# 3x4 matrix
Y = [[5,8,1,2],
    [6,7,3,0],
    [4,5,9,1]]
# result is 3x4
result = [[0,0,0,0],
         [0,0,0,0],
         [0,0,0,0]]

# iterate through rows of X
for i in range(len(X)):
   # iterate through columns of Y
   for j in range(len(Y[0])):
       # iterate through rows of Y
       for k in range(len(Y)):
           result[i][j] += X[i][k] * Y[k][j]

for r in result:
   print(r)
    def area(self):
        return self.radius**2*3.14
```

```
    def perimeter(self):
        return 2*self.radius*3.14

NewCircle = Circle(8)
print(NewCircle.area())
print(NewCircle.perimeter())
```
****
output..

200.96
50.24

Q2 Write a Python program to check if a nXn matrix is symmetric.

Program:

```
def Symmetric(a, n):
    for i in range(n):
        for j in range(n):
            if (a[i][j] != a[j][i]):
                return False
        return True
a = [ [ 1, 3, 5 ], [ 3, 2, 4 ], [ 5, 4, 9 ] ]
print("Given matrix: ")
print(a)
if (Symmetric(a, 3)):
    print("Given matrix is symmetric")
else:
    print("Given matrix is not a symmetrics")
```
****
output..

Given matrix:
[[1, 3, 5], [3, 2, 4], [5, 4, 9]]
Given matrix is symmetric

slip no 4

Q1) Program to check the number is Armstrong or not.

Program:

```
num = int(input("Enter a number: "))
sum = 0
temp = num
while temp > 0:
   digit = temp % 10
   sum += digit ** 3
   temp //= 10
if num == sum:
   print(num,"is an Armstrong number")
else:
   print(num,"is not an Armstrong number")
```

**********output**********

```
Enter a number: 589
589 is not an Armstrong number
```

Q2) Write a Python function to calculate the sum of digits of a number. Use this function in main to accept a number and print sum of its digits.

Program:

```
def getSum(n):

    sum = 0
    for digit in str(n):
      sum += int(digit)
    return sum

n = 12345
```

```
print(getSum(n))
```

**********output**********

15

Slip no-5

Q1) Write a Python program to multiply two matrices. Write separate functions to accept, display and multiply the matrices.

Program:

```
# Program to multiply two matrices using nested loops
# 3x3 matrix
X = [[12,7,3],
    [4 ,5,6],
    [7 ,8,9]]
# 3x4 matrix
Y = [[5,8,1,2],
    [6,7,3,0],
    [4,5,9,1]]
# result is 3x4
result = [[0,0,0,0],
          [0,0,0,0],
          [0,0,0,0]]

# iterate through rows of X
for i in range(len(X)):
   # iterate through columns of Y
   for j in range(len(Y[0])):
       # iterate through rows of Y
       for k in range(len(Y)):
           result[i][j] += X[i][k] * Y[k][j]

for r in result:
   print(r)
```

**********output**********

```
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
```

Q2) Write a Python program to accept real number x and integer n and calculate the sum of first n terms of the series x+ x/3!+ x/5!+ x/7!+…

program:

Slip no -6

Q1) Write a Python program to accept a matrix of size m x n and display transpose of a given matrix.

Program:

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

# initializing another (2 x 3) matrix to store the result.
transpose = [[0, 0, 0], [0, 0, 0]]

# iterating the rows and then columns of each row
for i in range(len(matrix)):
    for j in range(len(matrix[0])):
        transpose[j][i] = matrix[i][j]

print(transpose)
```

**********output**********

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

Q2) Write Python program to find the maximum number from an array of n integers.

Program:

Slip no-7

Q1 Write a Python program to add two matrices of order mXn

program:

```
X = [[12,7,3],
    [4 ,5,6],
    [7 ,8,9]]

Y = [[5,8,1],
    [6,7,3],
    [4,5,9]]

result = [[0,0,0],
          [0,0,0],
          [0,0,0]]
```

```
for i in range(len(X)):
   # iterate through columns
   for j in range(len(X[0])):
      result[i][j] = X[i][j] + Y[i][j]

for r in result:
   print(r)
```

**********output**********

```
[17, 15, 4]
[10, 12, 9]
[11, 13, 18]
```

Q2 Write a menu driven program to perform the following operations on an integer. Write separate functions.
1. Check if is even or odd
2. Check if it is prime
3. Exit

program:

```
while True:
    print("\nMAIN MENU")
    print("1. Calculate evev or odd")
    print("2. Calculate prime")
    print("3. Exit")
    choice1 = int(input("Enter the Choice:"))

    if choice1==1:
        print("\nCALCULATE even and odd")
        num = int(input("Enter a number: "))
        if (num % 2) == 0:
            print("{0} is Even".format(num))
        else:
```

```
        print("{0} is Odd".format(num))

    if choice1 == 2:
        num = 21
        flag = False

        if num > 1:
            for i in range(2, num):
                if (num % i) == 0:

                    flag = True
                    break

        if flag:
            print(num, "is not a prime number")
        else:
            print(num, "is a prime number")

    if choice1 ==3:
        breakS
```

**********output**********

MAIN MENU

```
1. Calculate evev or odd
2. Calculate prime
3. Exit
Enter the Choice:1

CALCULATE even and odd
Enter a number: 3
3 is Odd

MAIN MENU
1. Calculate evev or odd
2. Calculate prime
3. Exit
Enter the Choice:1

CALCULATE even and odd
Enter a number: 4
4 is Even

MAIN MENU
1. Calculate evev or odd
2. Calculate prime
3. Exit
Enter the Choice:2
21 is not a prime number
```

```
MAIN MENU
1. Calculate evev or odd
2. Calculate prime
3. Exit
```

Slip no-8

Q1) Write a Python program to check if a matrix is upper triangular program:

```
def isuppertriangular(M):
    for i in range(1, len(M)):
        for j in range(0, i):
            if(M[i][j] != 0):
                return False
    return True

# Driver function.
M = [[1,3,5,3],
    [0,4,6,2],
    [0,0,2,5],
    [0,0,0,6]]

if isuppertriangular(M):
    print ("Yes")
else:
    print ("No")
```

**********output**********

Yes

Q2) Accept two numbers and perform the following operations till the user selects Exit.

i. Maximum
ii. Display all numbers between the two
iii. Sum and average
iv EXIT

program:

```
a=int(input("Enter the no 1:"))
b=int(input("Enter the no 2:"))
while True:
    print("\nMAIN MENU")
    print("1. Print maximum")
    print("2. display bet two")
    print("3. Sum and average")
    print("4. Exit")
    choice1 = int(input("Enter the Choice:"))



    if choice1==1:

        if (a > b):
            largest = a

        elif (b > a):
            largest = b

        else:
            print("both no are equal")

        print(largest)

    if choice1 == 2:
```

```
      for num in range(a, b + 1):
       print(num)

   if choice1 ==3:
      l=0
      count = 0
      for num in range(a, b + 1):

         l=l+1
         count += num



      avg = count/l

      print("sum = ", count)
      print("average = ", avg)

   if choice1==4:
      break
```

**********output**********

```
Enter the no 1:5
Enter the no 2:10

MAIN MENU
1. Print maximum
2. display bet two
3. Sum and average
4. Exit
Enter the Choice:1
10

MAIN MENU
1. Print maximum
2. display bet two
3. Sum and average
4. Exit
Enter the Choice:2
5
6
7
8
9
10

MAIN MENU
1. Print maximum
```

```
2. display bet two
3. Sum and average
4. Exit
Enter the Choice:3
sum =  45
average =  7.5

MAIN MENU
1. Print maximum
2. display bet two
3. Sum and average
4. Exit
Enter the Choice:4
```

Slip no -9

Q1) Write Python program to subtract two matrices of order mXn [15 marks]

```
matrix1 = [[10, 11, 12],
          [13, 14, 15],
          [16, 17, 18]]
matrix2 = [[1, 2, 3],
          [4, 5, 6],
          [7, 8, 9]]
rmatrix = [[0, 0, 0],
          [0, 0, 0],
          [0, 0, 0]]

for i in range(len(matrix1)):
   for j in range(len(matrix1[0])):
       rmatrix[i][j] = matrix1[i][j] - matrix2[i][j]

for r in rmatrix:
   print(r)
```

****

output

```
[9, 9, 9]
[9, 9, 9]
[9, 9, 9]
```

Q2) Write a Python program to accept n integers in an array and search for a specific number

program:

```
from array import *
a=array('i', [])
n=int(input("Enter the length of the array: "))
for i in range(n):
    x=int(input("Enter a value: "))
    a.append(x)
print(a)

s=int(input("Enter the number to be searched: "))
c=0
for j in range(n):
    if a[j] == s:
        print("FOUND!")
        c+= 1
        break

if(c!=1):
    print("NOT FOUND!")
```

****

output

```
    Enter the length of the array: 3
Enter a value: 13
Enter a value: 56
Enter a value: 789
array('i', [13, 56, 789])
Enter the number to be searched: 56
FOUND!
```

Slip no-10

Q1) Write a function in Python to reverse an integer. Use this in main.

Program:

```
num = int(input("Enter the number: "))
    revr_num = 0    # initial value is 0. It will hold the reversed number
    def recur_reverse(num):
        global revr_num   # We can use it out of the function
        if (num > 0):
            Reminder = num % 10
            revr_num = (revr_num * 10) + Reminder
            recur_reverse(num // 10)
        return revr_num


    revr_num = recur_reverse(num)
    print("n Reverse of entered number is = %d" % revr_num)
```

**********output**********

```
Enter the number: 1234
n Reverse of entered number is = 4321
```

Slip no-13

Q1) Write a Python program to calculate occurrences of a number in an array of n integers

program:

```
def countX(lst, x):
    count = 0
    for ele in lst:
        if (ele == x):
            count = count + 1
    return count


# Driver Code
lst = [8, 6, 8, 10, 8, 20, 10, 8, 8]
x = 8
print('{} has occurred {} times'.format(x,countX(lst, x)))
```

**********output**********

8 has occurred 5 times

Slip no-15

Q1) Write a Python program to accept an array of n integers and find the maximum and minimum

program:

Slip no-16

Q1) Write a function in Python to check if a number is prime. Use this function to display the first 20 prime numbers.

Program:

```
lower = 1
upper = 20

print("Prime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 1):
   # all prime numbers are greater than 1
   if num > 1:
       for i in range(2, num):
           if (num % i) == 0:
               break
       else:
           print(num)
```

**********output**********

```
Prime numbers between 1 and 20 are:
2
3
5
7
11
13
17
19
```

Q2) Write a Python program to add two matrices of order mXn

Program to add two matrices using nested loop

program:

```
X = [[1,2,3],
         [4 ,5,6],
```

```
        [7 ,8,9]]

Y = [[9,8,7],
        [6,5,4],
        [3,2,1]]


result = [[0,0,0],
        [0,0,0],
        [0,0,0]]

# iterate through rows
for i in range(len(X)):
# iterate through columns
        for j in range(len(X[0])):
                result[i][j] = X[i][j] + Y[i][j]

for r in result:
        print(r)
**********output**********
[10, 10, 10]
[10, 10, 10]
[10, 10, 10]
```

Slip no-17

Q1) Write a Python program to display n lines of the following pattern. [15 marks]

```
1
2 3
4 5 6
```

program:

```
# Function to demonstrate printing pattern of numbers
def contnum(n):

    # initializing starting number
    num = 1

    # outer loop to handle number of rows
    for i in range(0, n):

        # not re assigning num
        # num = 1

        # inner loop to handle number of columns
        # values changing acc. to outer loop
        for j in range(0, i+1):

            # printing number
            print(num, end=" ")

            # incrementing number at each column
            num = num + 1

        # ending line after each row
        print("\r")

n = 3
```

```
# sending 5 as argument
# calling Function
contnum(n)
```

**********output**********

```
1
2 3
4 5 6
```

Slip no-18

Q1)Write a function in Python to calculate sum of digits of an integer. Use this function in main

program:

```
# Function to get sum of digits
def getSum(n):

    sum = 0
    for digit in str(n):
      sum += int(digit)
    return sum

n = 12345
print(getSum(n))
```

**********output**********

15

Q2)Write a Python program to accept n integers in an array and count the frequency of each element of an array.

Program:

```
#Initialize array
arr = [1, 2, 8, 3, 2, 2, 2, 5, 1];
#Array fr will store frequencies of element
fr = [None] * len(arr);
visited = -1;

for i in range(0, len(arr)):
    count = 1;
    for j in range(i+1, len(arr)):
        if(arr[i] == arr[j]):
            count = count + 1;
            #To avoid counting same element again
            fr[j] = visited;

    if(fr[i] != visited):
        fr[i] = count;
print("---------------------");
print(" Element | Frequency");
print("---------------------");
for i in range(0, len(fr)):
    if(fr[i] != visited):
        print("    " + str(arr[i]) + "    |    " + str(fr[i]));
print("---------------------");
```

**********output**********

```
--------------------
 Element | Frequency
--------------------
   1    |    2
```

2 | 4
8 | 1
3 | 1
5 | 1

Slip no-20

Q1)Write a Python program to accept two matrices of size m x n and calculate Addition of Matrices.

Program :

```
X = [[1,2,3],
     [4 ,5,6],
     [7 ,8,9]]

Y = [[9,8,7],
     [6,5,4],
     [3,2,1]]


result = [[0,0,0],
          [0,0,0],
          [0,0,0]]

# iterate through rows
for i in range(len(X)):
# iterate through columns
     for j in range(len(X[0])):
          result[i][j] = X[i][j] + Y[i][j]

for r in result:
     print(r)
```

**********output**********

```
[10, 10, 10]
[10, 10, 10]
[10, 10, 10]
```