Dr. D. Y. Patil Arts, Commerce & Science College

INDEX

Name of Student:
Roll No.:
Batch:
Practical - 1

Q.1) Find the data type of the following.

1) >>> A=36
   >>> type(A)
   O/P:
   >>> <class 'int'>

2) >>> A=3+2j
   >>> type(A)
   O/P:
   >>> <class 'complex'>

3) >>> A=3.14
   >>> type(A)
   O/P:
   >>> <class 'float'>

4) >>> A='python'
   >>> type(A)
   O/P:
   >>> <class 'str'>

5) >>> A=[1,2,3,4,5]
   >>> type(A)
   O/P:
   >>> <class 'list'>

Q.2) Print the following string using print statement.

1) >>> A='Hello World !'
   >>> print(A)
   O/P:
   >>> Hello World !

2) >>> B='Mathematics'
   >>> Print(B)

```
   O/P:
   >>> Mathematics
```

Q.3) Evaluate the following expression.

```
1) >>> A=10%2+7-(3+7)*10/2
   >>> print(A)
   O/P:
   >>> -43.0

2) >>> B=3*10//3+10%3
   >>> print(B)
   O/P:
   >>> 11
```

Q.4) Compare the following expression.

```
1) >>> A=21
   >>> B=15
   >>> print(A=B)
   O/P:
   >>> False

2) >>> A=4
   >>> B=5
   >>> print(A!=B)
   O/P:
   >>> True

3) >>> A=10
   >>> B=15
   >>> print(A<=B)
   O/P:
   >>> True

4) >>> A=4
   >>> B=8
   >>> print(A<B)
   O/P:
   >>> True
```

Q.5) Print the following terms by importing math and c math module.

1) log(10) with base 10

```
   I/P:
   >>> import math
   >>> print(math.log10(10)
   O/P:
   >>> 1.0

2) sin(30)
   I/P:
   >>> Import math
   >>> print(math.sin(30))
   O/P:
   >>> -0.9880316240928618

3) Value of pi
   I/P:
   >>> import math
   >>> print(math.pi)
   O/P:
   >>> 3.141592653589793

4) tan(45)
   I/P:
   >>> import math
   >>> print(math.tan(45))
   O/P:
   >>> 1.6197751905438615

5) 3+5j
   I/P:
   >>> a = 3+5j
   >>> print(a)
   O/P:
   >>> (3+5j)

Q.6) Accept the different value by using input() statement and print it
     from the following code.
1) >>> A=input("A:")
   >>> A:30
   >>> print(A)
   >>> 30
```

2)
```
>>> Name = input (“Enter the Name:  ”)
>>> Akash
>>> Age = int(input(“Age:  ”))
>>> 20
>>> City = input(“City Name:  ”)
>>> Pune
```

Name of Student:
Roll No.:
Batch:
Practical - 2

Q.1) Define string x = "Math is Easy" & y = "To All"
a) Find length of x and y .
b) Pick character with index 3 and -1 in x and 2 & 4 in y .
c) Using operator + & * find 2x+y , 3*x .

Ans:

a)
```
>>> x = "Math is Easy"
>>> y = "To All"
>>> len(x)
>>> 12
>>> len(y)
>>> 6
```

b)
```
>>> x = "Math is Easy"
>>> y = "To All"
>>> x[3]
>>> 'h'
>>> x[-1]
>>> 'y'
>>> y[2]
>>> ' '
>>> y[4]
>>> 'l'
```

c)
```
>>> 2*x + y
>>> 'Math is EasyMath is EasyTo All'
>>> 3*x
>>> 'Math is EasyMath is EasyMath is Easy'
```

Q.2) Write a python program of concatenation and repetition of string and list.

Ans:
```
# concatenation of string :-
>>> x = "hello "
>>> y = "I am in sybsc(cs)"
>>> x + y
```

```
>>> 'hello I am in sybsc(cs)'

# repetition of string :-
>>> x = "hello "
>>> y = "I am in sybsc(cs)"
>>> x*2
>>> 'hellohello'
>>> y*2
>>> 'I am in sybsc(cs)I am in sybsc(cs)'

# concatenation of list :-
>>> L1 = ['mango','apple','banana',45,78,'kiwi']
>>> L2 = ['bottle','book',66,90,'pen']
>>> L1 + L2
>>> ['mango', 'apple', 'banana', 45, 78, 'kiwi', 'bottle', 'book',
     66, 90, 'pen']

# repetition of list :-
>>> L1 = ['mango','apple','banana',45,78,'kiwi']
>>> L2 = ['bottle','book',66,90,'pen']
>>> L1 * 2
>>> ['mango', 'apple', 'banana', 45, 78, 'kiwi', 'mango', 'apple',
    'banana', 45, 78, 'kiwi']
>>> L2 * 2
>>> ['bottle', 'book', 66, 90, 'pen', 'bottle', 'book', 66, 90,
    'pen']
```

Q.3) Write a python program to create a nested list & diaplay its element.

Ans:

```
>>> L = ['a', 'b', ['cc', 'dd', ['eee', 'fff']], 'g', 'h']
>>> print(L)
>>> ['a', 'b', ['cc', 'dd', ['eee', 'fff']], 'g', 'h']
>>> Print(L[2])
>>> [‘cc’,’dd’,[‘eee’,’fff’]]
```

Q.4) Write a python program to create list using range function.

Ans:

```
>>> My_list = [*range(10, 20, 1)]
>>> print(My_list)
>>> [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

Q.5) Write a python program using membership operator ('not in' and 'is' operator) for list.
Use: x=[1,2,3,'Rose','Pink','math','easy']

Ans:

```
# 'in' operator:

L1=[1,2,3,'Rose','Pink','math','easy']
L2=[2,3,4,'red']
if L1 in L2:
      print("True")
else:
      print("False")

OUTPUT-
>>> False

# 'not in' operator:
L1=[1,2,3,'Rose','Pink','math','easy']
L2=[2,3,4,'red']
if L1 not in L2:
      print("True")
else:
      print("False")

OUTPUT-
>>> True
```

Q.6) Define tuple with single element & verify its type.

Ans:

```
>>> tuple1 = (5) type(tuple1)
>>> <class 'int'>
```

Q.7) Write a python program to write a function that return area and circumference of Circle.

Ans:

```
from math import *
def f(r):
      c = 2 * pi * r
      a = pi * r * r
      return c, a,
print(f(5))
```

```
OUTPUT-
(31.41592653589793, 78.53981633974483)
```

Q.8) Construct a valid list in python to convert it into a string.

Ans:

```
>>> mylist = ['I', 'me', 'you', 'they']
>>> mystr = ','.join(mylist)
>>> print(mystr)
O/P
>>> I,me,you,they
>>> type(mystr)
O/P
>>> <class 'str'>
>>> type(mylist)
O/P
>>> <class 'list'>
```

Q.9) Write a python program using range() function to display all prime Numbers between 1 to 100.

Program:

```
Lower = 2
Upper = 100
print("Prime numbers between", Lower, "to", Upper, "are :")
for n in range(Lower, Upper + 1):
      for i in range(2, n):
            if n % i == 0:
                  break
      else:
            print(n)
```

```
OUTPUT:
Prime numbers between 2 to 100 are :
2
3
5
7
11
13
17
19
23
```

```
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

Q.10) Write a string traversal program with while loop.

Program:

```
Z = “Black Bear”
Index = 0
While index < len(Z):
      Letter = Z[index]
      print(Letter)
      Index = Index + 1

OUTPUT:
B
l
a
c
k

B
e
a
r
```

Name of Student:
Roll No.:
Batch:
Practical - 3

Q.1) Find the range of the following function

a) range(10)
b) range(1,5)
c) range(1, 10, 2)
d) range(5,2)
e) range(10, -10, -2)

Ans:

a)

```
L1 = range(10)
for i in L1:
      print(i, end='  ')
print()
```

OUTPUT-

```
0  1  2  3  4  5  6  7  8  9
```

b)

```
L1 = range(1, 5)
for i in L1:
      print(i, end='  ')
print()
```

OUTPUT-

```
1  2  3  4
```

c)

```
L1 = range(1, 10, 2)
for i in L1:
      print(i, end='  ')
print()
```

OUTPUT-

```
1  3  5  7  9
```

d)

```
L1 = range(2, 20)
for i in L1:
      print(i, end='  ')
print()
```

OUTPUT-

```
2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19
```

e)

```
L1 = range(-10, 10, 2)
for i in L1:
      print(i, end='  ')
print()
```

OUTPUT-

```
-10  -8  -6  -4  -2  0  2  4  6  8
```

Q.2) Write a python program to find gcd of two numbers using for loop.

Ans:

```
import math
def phi(n):
      for x in range(1, n):
            if math.gcd(n,x)==1:
                  print(x, end=' ')
phi(10)
```

OUTPUT-

```
1 3 7 9
```

Q.3) Write a python program to find table of 22 using for loop.

Ans:

```
def table(n):
      for i in range(1,11):
            print (n*i, end= ' ')
table(22)
```

OUTPUT-

```
22 44 66 88 110 132 154 176 198 220
```

Q.4) Write a python program to find table of 19 using while loop.

Ans:

```
i=1
n=int(input("Table of  "))
while i<=10:
      print(n*i)
      i=i+1
Table of 19

OUTPUT-
19
38
57
76
95
114
133
152
171
190
```

Q.5) Write a python program to find first 10 terms of Fibonacci sequence using for loop.

Ans:

```
a = 0
b = 1
for i in range(10):
      a, b=b, a+b
      print(a, end= ' ')

OUTPUT-
1 1 2 3 5 8 13 21 34 55
```

Q.6) Write a python program to check the student is pass, fail or distinction.

Ans:

```
ame = input("Name of the student : ")
Marks = int(input('Marks =  '))
if Marks > 39:
```

```
      if Marks > 75:
            print("first class with distinction")
      elif Marks < 76 and Marks > 59:
            print("first class")
      else:
            print("Second class")
else:
      print("fail")
```

OUTPUT-

```
Name of the student :  Nisha
Marks =  71
first class
```

Name of Student:
Roll No.:
Batch:
Practical - 4

Q.1) Write a program to find divisors of any number using input function.

1) Write a python programme to find divisors of any number using input function.

Ans-

```
n = int(input("Divisor of : "))
for i in range(1,n+1):
      if  n%i == 0:
            print(i)
```

OUTPUT-

```
>>> Divisor of :10
    1
    2
    5
    10
```

Q.2) Write a python program to print the absulute value of given number.

Ans-

```
n = float(input("Enter the value:"))
absolute = abs(int(n))
print("\n The absolute value of the number is:->",absolute)
```

OUTPUT-

```
>>>  Enter the value: 5.23
     The absolute value of the number is:-> 5
```

Q.3) Write a python program to print the table of given number.

Ans-

```
i = 1
n = int(input("Table of :"))
while i<=10:
      print(n*i) i=i+1
```

OUTPUT-

```
Table of : 5
5
10
15
20
25
30
35
40
45
50
```

Q.4) Define a function that print all integers between 1 to n that are relatively prime.

Ans-

```
import math
def phi(n):
      for x in range (1,n):
            if math.gcd(n,x)==1:
                  print(x)
```

OUTPUT-

```
phi(10)
1
3
7
9
```

Q.5) Define Euler's phi function in python.

Ans-

```
Import math
def phi(n):
      i =0:
            for i in range(1,n):
                  if math.gcd(n,i) == 1:
                        i = i+1
                        print(I, end=' ')
```

OUTPUT-

```
phi(10)
2 4 8 10
```

Q.6) Write a program to check the person is eligible for voting or not and senior citizen or not.

Ans-

```
a = int(input("Enter your age :"))
if a>=60:
      print("You are eligible for Voting and you are senior
            citizen ")
elif 18<=a<60:
      print("You are eligible for voting and you are not senior
            citizen")
else:
      print("You are not eligible for voting")
```

OUTPUT-

```
Enter your age :5
You are not eligible for voting

Enter your age : 55
You are eligible for voting and you are not senior citizen

Enter your age : 70
You are eligible for Voting and you are senior citizen
```

Name of Student:

Roll No.:

Batch:

Practical – 5

Q.1) For vectors $U = \begin{bmatrix} 5 \\ 6 \\ 0 \end{bmatrix}$ & $V = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}$

Find a) U+V b) 3V c) 2U + 3V

Ans:

```
from sympy import *
U = Matrix([[2],[5],[-3]])
V = Matrix([[1],[0],[-2]])
```

a)
```
>>> U + V
   Matrix([
[ 3],
[ 5],
[-5]])
```

b)
```
>>> 3 * V
   Matrix([
[ 3],
[ 0],
[-6]])
```

c)
```
>>> 2*U+3*V
   Matrix([
[  7],
[ 10],
[-12]]
```

Q.2) Construct the following matrices using python

a) Identity matrix of order 6

b) Zero matirx with order 5x6

c) ones matrix of order 5x4

d) Diagonal matrix with (4,-5,1)vas a diagonal element.

Ans:

a)
```
from sympy import *
eye(6)
```

```
Matrix([
[1, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0],
[0, 0, 1, 0, 0, 0],
[0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 0, 1]])
```

b)
```
>>> zeros(5,6)
Matrix([
[0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0]])
```

c)
```
>>> ones(5,4)
Matrix([
[1, 1, 1, 1],
[1, 1, 1, 1],
[1, 1, 1, 1],
[1, 1, 1, 1],
[1, 1, 1, 1]])
```

d)
```
>>> diag(4,-5,1)
Matrix([
[4,  0, 0],
[0, -5, 0],
[0,  0, 1]])
```

Q.3) For the following matrices

$$A = \begin{bmatrix} 4 & 2 & 4 \\ 4 & -1 & 1 \\ 2 & 4 & 2 \end{bmatrix} \quad \& \quad B = \begin{bmatrix} 5 & 2 & 3 \\ 3 & -7 & 5 \\ 3 & 1 & -1 \end{bmatrix}$$

Find a) A+B b) A-B c) A^{-1} d) B*A e) $B^{-1}AB$
f) A^4

Ans:

```
from sympy import *
A = Matrix([[4,2,4],[4,-1,1],[2,4,2]])
```

```
B = Matrix([[5,2,3],[3,-7,5],[3,1,-1]])
```

a)
```
>>> A + B
Matrix([
[9,  4, 7],
[7, -8, 6],
[5,  5, 1]])
```

b)
```
>>> A - B
Matrix([
[-1, 0,  1],
[ 1, 6, -4],
[-1, 3,  3]])
```

c)
```
>>> A.inv()
Matrix([
[-1/6,  1/3,  1/6],
[-1/6, 0, 1/3],
[ 1/2, -1/3, -1/3]])
```

d)
```
>>> B * A
Matrix([
[34, 20, 28],
[-6, 33, 15],
[14,  1, 11]])
```

e)
```
>>> B.inv()*A*B
Matrix([
[ 522/59, -303/59, 405/59],
[ -22/59, 46/59, -72/59],
[-108/59, 435/59, -273/59]])
```

f)
```
>>> A**4
Matrix([
[2060, 1198, 1622],
[1106,  613,  857],
[1456,  848, 1120]])
```

Q.4) For matrix $A = \begin{bmatrix} -5 & 2 & 3 \\ 3 & -7 & 5 \\ -3 & 10 & -11 \end{bmatrix}$. Do the following with sequence,

a) Delete 2nd column

b) Add row [2,3,0] in first row

c) Delete last row

Ans:

```
from sympy import*
A = Matrix([[-5,2,3],[3,-7,5],[-3,10,-11]])
>>> A
Matrix([
[-5,  2, 3],
[ 3, -7, 5],
[-3, 10, -11]])
```

a) `>>> A.col_del(1) A`

```
Matrix([
[-5,  3],
[ 3,  5],
[-3, -11]])
```

b) `>>> A.row_insert(0, Matrix([[2, 3, 0]]))`

```
Matrix([
[ 2, 3, 0],
[-5, 2, 3],
[3, -7,  5],
[-3, 10, 11]])
```

c) `>>> A.row_del(2) A`

```
Matrix([
[-5,  2, 3],
[ 3, -7, 5]])
```

Q.5) Using Python apply the following operations on $A = \begin{bmatrix} 1 & 7 & 1 \\ 4 & -2 & 1 \\ 3 & 1 & 2 \end{bmatrix}$

a) Delete 3rd row

b) Delete 1st column

c) Delete the last column

Ans:

```
from sympy import*
A = Matrix([[1,7,1],[4,-2,1],[3,1,2]])
>>> A
```

```
Matrix([
[1,  7, 1],
[4, -2, 1],
[3,  1, 2]])
```

a)
```
>>> A.row_del(2)A
Matrix([
[1,  7, 1],
[4, -2, 1]])
```

b)
```
>>> A.col_del(0)A
Matrix([
[ 7, 1],
[-2, 1]])
```

c)
```
>>> A.col_del(1)A
Matrix([
[ 7],
[-2]])
```

Q. 6) For matrix $A = \begin{bmatrix} 1 & 0 & 4 \\ 2 & 1 & -1 \\ 3 & 4 & 2 \end{bmatrix}$ insert $A = \begin{bmatrix} 5 \\ 3 \\ 0 \end{bmatrix}$ as a 3rd column.

Ans:

```
from sympy import*
A = Matrix([[1,0,4],[2,1,-1],[3,4,2]])
>>> A
Matrix([
[1, 0,  4],
[2, 1, -1],
[3, 4,  2]])

>>> A.col_insert(2,Matrix([[5],[6],[0]]))
Matrix([
[1, 0, 5,  4],
[2, 1, 6, -1],
[3, 4, 0,  2]])
```

Name of Student:
Roll No.:
Batch:
Practical - 6

Q.1) For matrix
D = [10, 2, 3]
[12, -7, 15]
[-15, 10, -11]
a) Find its transpose
b) Find its determinant
c) Find invers if it exist.

Ans-

```
from sympy import *
D = Matrix([[10, 2, 3], [12, -7, 15], [-15, 10, -11]])
>>> D
Matrix([
[ 10, 2, 3],
[ 12, -7, 15],
[-15, 10, -11]])
a) >>> D.T                # Transpose of matrix D
   Matrix([
   [10, 12, -15],
   [ 2, -7, 10],
   [ 3, 15, -11]])
b) >>> D.det()            # determinant of matrix D
   -871
c) >>> D.inv()            # inverse of matrix D
   Matrix([
   [ 73/871, -4/67, -51/871],
   [93/871, 5/67, 114/871],
   [-15/871, 10/67, 94/871]])
```

Q.2) For matrix
B=[1 -1 -2 4]
[2 -1 -1 2]
[2 1 4 16]
a) Find reduce row echelon from
b) Find column space
c) Find null space

Ans-

```
from sympy import *
B = Matrix([[1,-1,-2,4],[2,-1,-1,2],[2,1,4,16]])
>>> B
Matrix([
[1, -1, -2, 4],
[2, -1, -1, 2],
[2, 1, 4, 16]])
```

a)
```
>>> B.rref()            # reduced row echelon form of matrix B
(Matrix([
[1, 0, 0, 24],
[0, 1, 0, 72],
[0, 0, 1, -26]]), (0, 1, 2))
```

b)
```
>>> B.columnspace()     # column space of matrix B
[Matrix([
[1],
[2],
[2]]), Matrix([
[-1],
[-1],
[ 1]]), Matrix([
[-2],
[-1],
[ 4]])]
```

c)
```
>>> B.nullspace()       # nullspace of matrix B
[Matrix([
[-24],
[-72],
[ 26],
[ 1]])]
```

Q.3) For matrix

```
A = [-5, 2, 3]
[-3, 10, -11]
[3,  -7, 5]
```

find rank of A.

Ans-

```
from sympy import *
```

```
A = Matrix([[-5,2,3],[3,-7,5],[-3,10,-11]])
>>> A
Matrix([
[-5, 2, 3],
[ 3, -7, 5],
[-3, 10, -11]])
>>> A.rank()            # rank of matrix A
3
```

Q.4) Find rank of matrix

```
M = [2 0 3]
[0 1 2]
[3 0 4]
```

Ans-

```
from sympy import *
M = Matrix([[2,0,3],[0,1,2],[3,0,4]])
>>> M
Matrix([
[2, 0, 3],
[0, 1, 2],
[3, 0, 4]])
>>> M.rank()            # rank of matrix M
3
```

Name of Student:

Roll No.:

Batch:

Practical – 7

Q.1) Find the solution of AX = b by Gauss elimination method where

$A = \begin{pmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ 2 & -1 & 2 \end{pmatrix}$ and $b = \begin{pmatrix} 3 \\ 6 \\ 9 \end{pmatrix}$

Program:

```
from sympy import *
x, y, z = symbols("x, y, z")
A = Matrix([[3, 2, -1], [2, -2, 4], [2, -1, 2]])
B = Matrix([[3], [6], [9]])
print(linsolve((A, b),[x, y, z]))
```

OUTPUT:

```
{(6, -11, -7)}
```

Q.2) Find the solution of AX=b by Gauss Jordan method where

$A = \begin{pmatrix} 7 & 6 & -8 \\ 7 & -2 & 2 \\ 6 & -1 & 2 \end{pmatrix}$ and $b = \begin{pmatrix} 3 \\ 0 \\ 9 \end{pmatrix}$

Program:

```
from sympy import *
x, y, z = symbols("x, y, z")
A = Matrix([[7, 6, -8], [7, -2, 2], [6, -1, 2]])
B = Matrix([[3], [0], [9]])
sol,param = A.gauss_jordan_solve(b)
```

OUTPUT:

```
Matrix([
[7/11],
[106/11],
[163/22]])
```

Q.3) Find the solution of system AX=b where $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ and $b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

Program:

```
from sympy import *
x, y, z = symbols("x, y, z")
A = Matrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
B = Matrix([[1], [2], [3]])
print(linsolve((A, b), [x, y, z]))
```

OUTPUT:

```
{(z - 1/3, 2/3 - 2*z, z)}
```

Q.4) Find the solution by Gauss elimination method and Gauss Jordan method where $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ and $b = \begin{pmatrix} 5 \\ 7 \end{pmatrix}$

1) by Gauss elimination method

Program:

```
from sympy import *
x, y = symbols("x, y")
A = Matrix([[2, 1], [1, 2]])
B = Matrix([[5], [7]])
print(linsolve((A, b), [x, y]))
```

OUTPUT:

```
{(1, 3)}
```

2) Gauss Jordan method

Program:

```
from sympy import *
x, y = symbols("x, y")
A = Matrix([[2, 1], [1, 2]])
b = Matrix([[5], [7]])
sol,param=A.gauss_jordan_solve(b)
```

OUTPUT:

```
Matrix([
[1],
[3]])
```

Q.5) Using linsolve command find the solution of

$5x - 2y + 3z = 2$

$x + y + z = 1$

$x + 2y - 2z = -10$

Program:

```
from sympy import *
x, y, z = symbols("x, y, z")
A = Matrix([[5, -2, 3], [1, 1, 1], [1, 2, -2]])
b = Matrix([[2],[1],[-10]])
print(linsolve((A, b), [x, y, z]))
```

OUTPUT:

```
{(-44/23, -13/23, 80/23)}
```

Q.6) Find LU factorization of $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$

Program:

```
from sympy import *
A = Matrix([[1,2,3],[4,5,6],[7,8,9]])
L,U,_=A.LUdecomposition()
```

OUTPUT:

```
>>> L
Matrix([
[1, 0, 0],
[4, 1, 0],
[7, 2, 1]])
>>> U
Matrix([
[1, 2, 3],
[0, -3, -6],
[0, 0, 0]])
```

Q.7) Use LU factorization to solve the system

$$2x + 2y + z = 6$$
$$2x + y + 2z = 8$$
$$x + 2y + 2z = 7$$

Program:

```
from sympy import *
x, y, z = symbols("x, y, z")
AB = Matrix([[2, 2, 1, 6], [2, 1, 2, 8], [1, 2, 2, 7]])
print(solve_linear_system_LU(AB, [x, y, z]))
```

OUTPUT:

```
{x: 7/5, y: 2/5, z: 12/5}
```

Q.8) Solve the system by using Gauss elimination method, Gauss Jordan method and LU decomposition method.

a) $\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 4 \\ 2 & 5 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 6 \\ 7 \\ 8 \end{pmatrix}$

1) By Gauss Elimination Method

Program:

```
from sympy import *
x, y, z = symbols("x, y, z")
A = Matrix([[1, 2, 3], [2, 1, 4], [2, 5, 1]])
b = Matrix([[6], [7], [8]])
print(linsolve((A, b), [x, y, z]))
```

```
OUTPUT:
      {(1, 1, 1)}

2) By Gauss Jordan Method
Program:
      from sympy import *
      x, y, z = symbols("x, y, z")
      A = Matrix([[1, 2, 3], [2, 1, 4], [2, 5, 1]])
      b = Matrix([[6], [7], [8]])
      sol,param=A.gauss_jordan_solve(b)
OUTPUT:
      Matrix([
      [1],
      [1],
      [1]])

3) By LU Decomposition
Program:
      from sympy import *
      x, y, z = symbols("x, y, z")
      AB = Matrix([[1, 2, 3, 6], [2, 1, 4, 7], [2, 5, 1, 8]])
      print(solve_linear_system_LU(AB, [x, y, z]))
OUTPUT:
      {x: 1, y: 1, z: 1}
```

Q.8 b) $-x + 2y + 2z = -1$

$x + y + 2z = 2$

$5x + 2z = 8$

```
1) By Gauss Elimination Method
Program:
      from sympy import *
      x, y, z = symbols("x, y, z")
      A = Matrix([[-1, 2, 2], [1, 1, 2], [5, 0, 2]])
      b = Matrix([[-1], [2], [8]])
      print(linsolve((A, b), [x, y, z]))
OUTPUT:
      {(3/2, 0, 1/4)}

2) By Gauss Jordan Method
      from sympy import *
      x, y, z = symbols("x, y, z")
```

```
      A = Matrix([[-1, 2, 2], [1, 1, 2], [5, 0, 2]])
      b = Matrix([[-1], [2], [8]])
      sol,param=A.gauss_jordan_solve(b)
OUTPUT:
      Matrix([
      [3/2],
      [  0],
      [1/4]])

3) By LU Decomposition
      from sympy import *
      x, y, z = symbols("x, y, z")
      AB = Matrix([[-1, 2, 2, -1], [1, 1, 2, 2], [5, 0, 2, 8]])
      print(solve_linear_system_LU(AB, [x, y, z]))
OUTPUT:
      {x: 3/2, y: 0, z: 1/4}
```

Name of Student:

Roll No.:

Batch:

Practical – 8

Q.1) Using Sympy module of python, find the eigen values and the corresponding eigen vectors of the matrix A = $\begin{bmatrix} 4 & 2 & 2 \\ 2 & 4 & 2 \\ 2 & 2 & 4 \end{bmatrix}$.

Ans-

```
from sympy import *
A = Matrix([[4,2,2],[2,4,2],[2,2,4]])
>>> A.eigenvals()
O/P:
{8: 1, 2: 2}
>>> A.eigenvects()
O/p:
[(2, 2, [Matrix([
[-1],
[ 1],
[ 0]]), Matrix([
[-1],
[ 0],
[ 1]])]), (8, 1, [Matrix([
[1],
[1],
[1]])])]
```

Q.2) Using Sympy module of python, find the eigen values and the corresponding eigen vectors of the matrix A = $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$.

Ans-

```
from sympy import *
A = Matrix([[1,1,1],[1,1,1],[1,1,1]])
>>> A.eigenvals()
{3: 1, 0: 2}
>>> A.eigenvects()
[(0, 2, [Matrix([
[-1],
[ 1],
[ 0]]), Matrix([
[-1],
```

```
[ 0],
[ 1]])]), (3, 1, [Matrix([
[1],
[1],
[1]])])]
```

Q.3) Using python find the eigen values of the following matrix:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Ans-

```
from sympy import *
A = Matrix([[1,1,1],[0,1,1],[0,0,1]])
>>> A.eigenvals()
{1: 3}
```

Q.4) Using python find the eigen vectors of the following matrix:

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix}.$$

Ans-

```
from sympy import *
A = Matrix([[2,1,1],[0,2,1],[0,0,2]])
>>> A.eigenvects()
[(2, 3, [Matrix([
[1],
[0],
[0]])])]
```

Q.5) Using Sympy module of python, find the eigen values and the corresponding eigen vectors of the matrix $A = \begin{bmatrix} 3 & -2 \\ 6 & -4 \end{bmatrix}$.

Ans-

```
from sympy import *
A = Matrix([[3,-2],[6,-4]])
>>> A.eigenvals()
{-1: 1, 0: 1}
>>> A.eigenvects()
[(-1, 1, [Matrix([
[1/2],
[  1]])]), (0, 1, [Matrix([
[2/3],
[  1]])])]
```

Name of Student:

Roll No.:

Batch:

Practical – 9

Q.1) Write a python program to diagonalizes the matrix $\begin{bmatrix} 3 & -2 \\ 6 & -4 \end{bmatrix}$ and find matrix P and D.

Program:

```
>>> from sympy import *
>>> A = Matrix([[3,-2],[6,-4]])
>>> P, D = A.diagonalize()
>>> P
OUTPUT:
Matrix([
[1, 2],
[2, 3]])
>>> D
OUTPUT:
Matrix([
[-1, 0],
[ 0, 0]])
```

Q.2) Using python program check whether the matrix A= $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 3 \end{bmatrix}$ is diagonalizable or not.

Program:

```
>>> from sympy import *
>>> A = Matrix([[0,1,0],[0,0,1],[0,0,3]])
>>> A.is_diagonalizable()
OUTPUT:
False
```

Q.3) Find the eigenvalues of the following matrix and hence check whether it is diagonalizable or not.

Program:

```
>>> from sympy import *
>>> A = Matrix([[1,2,2],[2,1,2],[2,2,1]])
>>> A.eigenvals()
OUTPUT:
{5: 1, -1: 2}
>>> A.is_diagonalizable()
OUTPUT:
```

```
True
```

Q.4) Using python program check whether the matrix A= $\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$ is diagonalizable or not.

Program:

```
>>> from sympy import *
>>> A = Matrix([[0,1,0],[0,0,1],[0,0,3]])
>>> A.is_diagonalizable()
OUTPUT:
True
```

Q.5) Write a python program to diagonalize the matrix $\begin{bmatrix}1 & -1 & 1\\-1 & 1 & -1\\1 & -1 & 1\end{bmatrix}$ and find matrix P and D.

Program:

```
>>> from sympy import *
>>> A = Matrix([[1,-1,1],[-1,1,-1],[1,-1,1]])
>>> P, D = A.diagonalize()
>>> P
OUTPUT:
Matrix([
[1, -1,  1],
[1,  0, -1],
[0,  1,  1]])
>>> D
OUTPUT:
Matrix([
[0, 0, 0],
[0, 0, 0],
[0, 0, 3]])
```

Name of Student:
Roll No.:
Batch:
Practical – 10

Q.1) Write a python program to find the correct root of the equation $x^3 - 4x^2 - 11x + 30$ using Newton Raphson method correct upto 4 decimal places. Take $x_0 = 10$.

Program:

```
def nr(f, g, x0, e):
      step = 1
      condition = True
      while condition:
            if g(x0) == 0.0:
                  print("Divided by zero error.")
                  break
            x1 = x0 - (f(x0)/g(x0))
            print('Iteration = ', step, 'x1 =', x1, 'and f(x1)=', f(x1))
            x0 = x1
            step = step + 1
            condition = abs(f(x1)) > e
      print('Required root is :', x1)

from math import *
deff(x):
      return x**3 - 4*x**2 - 11*x + 30
defg(x):
      return 3*x**2 -8*x - 11
nr(f, g, 10, 0.00001)
```

OUTPUT:

```
Iteration = 1 x1= 7.511961722488039 and f(x1)= 145.54690711661283
Iteration = 2 x1= 6.029708528510192 and f(x1)= 37.46810047097503
Iteration = 3 x1= 5.2778576911468225 and f(x1)= 7.539290423642491
Iteration = 4 x1= 5.029400988117186 and f(x1)= 0.7151577286837636
Iteration = 5 x1= 5.000387815638643 and f(x1)= 0.009309229796429008
Iteration = 6 x1= 5.0000000689141375 and f(x1)= 1.653939342816102e-06
Required root is : 5.0000000689141375
```

Q.2) Using python programming find the correct root of the function $x^3 + 2x$, in [-10, 10] using Newton Raphson method correct upto 4 decimal places. Take

$x_0 = 3$

Program:

```
def newtonraphson(f,g,x0,e):
      step = 1
      condition = True
      while condition:
            if g(x0) == 0.0:
                  print('Divided by zero error!')
                  break
            x1 = x0 - (f(x0)/g(x0))
            print('Iteration =',step,'x1=',x1,'and f(x1)=',f(x1))
            x0 = x1
            step = step + 1
            condition = abs(f(x1))> e
      print('Required root is :',x1)
from math import*
def f(x):
      return x**3+2*x
def g(x):
      return 3*x*2+2
newtonraphson(f,g,3,0.00001)
```

```
OUTPUT:
Iteration = 1 x1= 1.35 and f(x1)= 5.160375
Iteration = 2 x1= 0.839071782178218 and f(x1)= 2.2688848832741244
Iteration = 3 x1= 0.5165318379707146 and f(x1)= 1.1708770250983709
Iteration = 4 x1= 0.286911684752419 and f(x1)= 0.5974414559037061
Iteration = 5 x1= 0.12637258636310406 and f(x1)= 0.25476334679541873
Iteration = 6 x1= 0.03400797677253169 and f(x1)= 0.06805528521500116
Iteration = 7 x1= 0.0031305686951099317 and f(x1)= 0.006261168071234248
Iteration = 8 x1= 2.911262335528837e-05 and f(x1)= 5.822524673525099e-05
Iteration = 9 x1= 2.5424001309411746e-09 and f(x1)= 5.084800261882349e-09
Required root is : 2.5424001309411746e-09
```

Q.3) Using python find the correct root of the function $e^x - \sin x$, in [0, 1] using Newton Raphson method correct upto 3 decimal places. Take $x_0 = 0.4$.

Program:

```
def newtonraphson(f,g,x0,e):
      step = 1
      condition = True
      while condition:
```

```
                if g(x0) == 0.0:
                        print('Divided by zero error!')
                        break
                x1 = x0 - (f(x0)/g(x0))
                print('Iteration =',step,'x1=',x1,'and f(x1)=',f(x1))
                x0 = x1
                step = step + 1
                condition = abs(f(x1))> e
        print('Required root is :',x1)
from math import*
def f(x):
        return e**x-sin(x)
def g(x):
        return e**x-cos(x)
newtonraphson(f,g,0.4,0.0001)

OUTPUT:
Iteration = 1 x1= -1.5314584096802757 and f(x1)= 1.2154464636832412
Iteration = 2 x1= -8.402567407072686 and f(x1)= 0.8534871583517203
Iteration = 3 x1= -10.038523764622981 and f(x1)= -0.575889990450599
Iteration = 4 x1= -9.334105736641906 and f(x1)= 0.09063639101087893
Iteration = 5 x1= -9.425107915627608 and f(x1)= -0.0002492819574764926
Iteration = 6 x1= -9.42485865376524 and f(x1)= 1.017479777190057e-11
Required root is : -9.42485865376524
```

Q.4) Using python find the correct root of the function e^x, in [-1, 1] using Regula Falsi method correct upto 4 decimal places.

Program:

```
def falseposition(f,x0,x1,e):
        if (f(x0)*f(x1))>0.0:
                print('given guess values do not break the root')
                print('try again with different guess values')
        else:
                step = 1
                condition = True
                while condition:
                        x2 = ((x0*f(x1))-(x1*f(x0)))/(f(x1)-f(x0))
                        print('Iteration =',step,'x2=',x2,'and f(x2)=',f(x2))
                        if (f(x0)*f(x2))<0:
                                x1=x2
                        else:
```

```
                        x0=x2
                    step = step + 1
                    condition = abs(f(x2))> e
            print('Required root is :',x2)
    from math import *
    def f(x):
    return e**x
    falseposition(f,-1,1,0.00001)
```

OUTPUT:

```
    given guess values do not break the root
    try again with different guess values
```

Q.5) Write a python program to estimate a root of an equation $\sin x = 8x$ in [-1, 2] using Regula Falsi method correct upto 4 decimal places.

Program:

```
    def falseposition(f,x0,x1,e):
        if (f(x0)*f(x1))>0.0:
            print('given guess values do not break the root')
            print('try again with different guess values')
        else:
            step = 1
            condition = True
            while condition:
                x2 = ((x0*f(x1))-(x1*f(x0)))/(f(x1)-f(x0))
                print('Iteration =',step,'x2=',x2,'and f(x2)=',f(x2))
                if (f(x0)*f(x2))<0:
                    x1=x2
                else:
                    x0=x2
                step = step + 1
                condition = abs(f(x2))> e
            print('Required root is :',x2)
    from math import *
    def f(x):
        return sin(x)-8*x
    falseposition(f,-1,2,0.00001)
```

OUTPUT:

```
    Iteration = 1 x2= -0.03477174210342762 and f(x2)= 0.24340920123557913
    Iteration = 2 x2= -0.002472371150056373 and f(x2)= 0.017306600569171028
```

```
Iteration = 3 x2= -0.00017848920441100228 and f(x2)= 0.001249424431824746
Iteration = 4 x2= -1.289949985729723e-05 and f(x2)= 9.029649900143835e-05
Iteration = 5 x2= -9.323246172555106e-07 and f(x2)= 6.526272320788709e-06
Required root is : -9.323246172555106e-07
```

Q.6) Write a python program to estimate a root of an equation $x^6 - x^4 - x^3 - 1 = 0$ in [1, 2] using Regula Falsi method correct upto 3 decimal places.

Program:

```
def falseposition(f,x0,x1,e):
    if (f(x0)*f(x1))>0.0:
        print('given guess values do not break the root')
        print('try again with different guess values')
    else:
        step = 1
        condition = True
        while condition:
            x2 = ((x0*f(x1))-(x1*f(x0)))/(f(x1)-f(x0))
            print('Iteration =',step,'x2=',x2,'and f(x2)=',f(x2))
            if (f(x0)*f(x2))<0:
                x1=x2
            else:
                x0=x2
            step = step + 1
            condition = abs(f(x2))> e
        print('Required root is :',x2)
from math import *
def f(x):
    return x**6-x**4-x**3-1
falseposition(f,1,2,0.0001)
```

OUTPUT:

```
Iteration = 1 x2= 1.048780487804878 and f(x2)= -2.0326812065849147
Iteration = 2 x2= 1.095902098406956 and f(x2)= -2.02625465045382
Iteration = 3 x2= 1.1405547871102417 and f(x2)= -1.9745687858492924
Iteration = 4 x2= 1.181971542449124 and f(x2)= -1.8763188793993664
Iteration = 5 x2= 1.2195209666846365 and f(x2)= -1.736023605208181
Iteration = 6 x2= 1.2527821911560477 and f(x2)= -1.563485345302133
Iteration = 7 x2= 1.2815830716506917 and f(x2)= -1.3718177813280636
Iteration = 8 x2= 1.3059945837122684 and f(x2)= -1.1747759787016694
Iteration = 9 x2= 1.3262884340768828 and f(x2)= -0.9843518672063674
Iteration = 10 x2= 1.3428741534122222 and f(x2)= -0.8093081791125121
Iteration = 11 x2= 1.3562332733435944 and f(x2)= -0.6547960779081032
```

```
Iteration = 12 x2= 1.3668634106635338 and f(x2)= -0.522787126619709
Iteration = 13 x2= 1.3752382162468697 and f(x2)= -0.4129045722984195
Iteration = 14 x2= 1.3817834582154178 and f(x2)= -0.32330643530603
Iteration = 15 x2= 1.386866280706462 and f(x2)= -0.25142151639549626
Iteration = 16 x2= 1.3907936546333808 and f(x2)= -0.1944716142576226
Iteration = 17 x2= 1.3938163600436084 and f(x2)= -0.14979341446752636
Iteration = 18 x2= 1.3961357162726984 and f(x2)= -0.11500652946132384
Iteration = 19 x2= 1.397911207105988 and f(x2)= -0.08807814104978506
Iteration = 20 x2= 1.399267908794765 and f(x2)= -0.06732573266722808
Iteration = 21 x2= 1.4003031659417187 and f(x2)= -0.05138733928068495
Iteration = 22 x2= 1.4010923011498884 and f(x2)= -0.039178108576233495
Iteration = 23 x2= 1.4016933402083296 and f(x2)= -0.029844100245050686
Iteration = 24 x2= 1.4021508343219684 and f(x2)= -0.022719021894551883
Iteration = 25 x2= 1.402498902027785 and f(x2)= -0.017286397718459412
Iteration = 26 x2= 1.4027636216577326 and f(x2)= -0.013147849509306475
Iteration = 27 x2= 1.4029648967266968 and f(x2)= -0.009997229582844547
Iteration = 28 x2= 1.4031179010184252 and f(x2)= -0.007599925145015174
Iteration = 29 x2= 1.403234192697005 and f(x2)= -0.005776522969814568
Iteration = 30 x2= 1.4033225701554424 and f(x2)= -0.004390041336661543
Iteration = 31 x2= 1.4033897276876817 and f(x2)= -0.0033360215325179965
Iteration = 32 x2= 1.4034407567769334 and f(x2)= -0.0025348791569839157
Iteration = 33 x2= 1.403479528759222 and f(x2)= -0.0019260231153448437
Iteration = 34 x2= 1.4035089865920407 and f(x2)= -0.001463347115464586
Iteration = 35 x2= 1.403531367121615 and f(x2)= -0.0011117810208047807
Iteration = 36 x2= 1.4035483702908902 and f(x2)= -0.0008446573178555461
Iteration = 37 x2= 1.4035612878889117 and f(x2)= -0.0006417025769653861
Iteration = 38 x2= 1.4035711014776593 and f(x2)= -0.0004875070439585016 4
Iteration = 39 x2= 1.403578556853421 and f(x2)= -0.0003703593861463794
Iteration = 40 x2= 1.4035842206529576 and f(x2)= -0.000281359952712279
Iteration = 41 x2= 1.4035885233787233 and f(x2)= -0.0002137462600102502
Iteration = 42 x2= 1.4035917920972842 and f(x2)= -0.00016238007107904906
Iteration = 43 x2= 1.4035942752871293 and f(x2)= -0.00012335744923852587
Iteration = 44 x2= 1.4035961617193395 and f(x2)= -9.371235608179873e-05
Required root is : 1.4035961617193395
```

Name of Student:
Roll No.:
Batch:
Practical – 11

Q.1 Write a python program to estimate the value of the integral $\int_0^1 1+x$ using trapezoidal rule, take h=0.2.

Program:

```
def trapezoidalf(f, a, b, n):
      h = float(b-a)/n
      result = f(a) + f(b)
      for i in range(1, n):
            result = result + 2 * f(a + i*h)
      result = (h/2)*result
      print(result)
      return(result)
from math import *
def f(x):
return 1 + x
trapezoidalf(f, 0, 1, 5)
```

OUTPUT:

```
1.5
```

Q.2 find the value of $\int_0^5 \frac{dx}{1+x^2}$ using trapezoidal rule, take h=0.5.

Program:

```
def trapezoidal(f,a,b,n):
      h = float(b-a)/n
      result = f(a) + f(b)
      for i in range(1,n):
            result=result+2*f(a+i*h)
      result=(h/2)*result
      print("Integration by Trap is: ", result)
      return(result)
from math import*
def f(x):
      return 1/(1+x**2)
trapezoidal(f,0,5,10)
```

OUTPUT:

```
trapezoidal(f,0,5,10)
Integration by Trap is: 17.5
```

Q.3 Solve the integral $\int_0^{10}(x+1)^3 dx$ using trapezoidal rule, take h=1.

Program:

```
def trapezoidal(f,a,b,n):
      h = float(b-a)/n
      result = f(a) + f(b)
      for i in range(1,n):
            result = result + 2*f(a + i*h)
      result = (h/2) * result
      print("Integration by trap is: ", result)
      return(result)
from math import *
def f(x):
      return (x+1)**3
trapezoidal(f,0,10,10)
```

OUTPUT:

```
trapezoidal(f,0,10,10)
Integration by trap is: 60.0
```

Q.4 Solve the integral $\int_{0.2}^{1.4}(\sin x - \log x + e^x)dx$ using trapezoidal rule, take h=0.2.

Program:

```
def trapezoidal(f,a,b,n):
      h = float(b-a)/n
      result = f(a) + f(b)
      for i in range(1,n):
            result = result + 2 * f(a + i*h)
      result = (h/2)*result
      print("Integration by trap is :", result)
      return(result)
from math import *
def f(x):
      return sin(x)-log(x)+exp(x)
trapezoidal(f,0.2,1.4,6)
```

OUTPUT:

```
trapezoidal(f,0.2,1.4,6)
Integration by trap is: 2.16
```

Q.5 Evaluate $\int_0^{2.5} e^x dx$ using trapezoidal rule, take h=0.5.

Program:

```
def trapezoidal(f, a, b, n):
      h = float(b-a)/n
```

```
        result = f(a) + f(b)
        for i in range(1,n):
        result = result +2*f(a + i*h)
        result = (h/2)*result
        print("Integration by trap is :", result)
        return(result)
  from math import *
  def f(x):
        return exp(x)
  trapezoidal(f,0,2.5,5)
```

OUTPUT:

```
  trapezoidal(f,0,2.5,5)
  Integration by trap is : 5.625
```

Name of Student:
Roll No.:
Batch:
Practical – 12

Q.1 Using Simpson's (1/3)rd rule, evaluate $\int_0^{10} \sqrt{1+x^2}dx$ with h=0.25.

Program:

```
def simpsons13(f,a,b,n):
      h = float(b-a)/n
      result = f(a) + f(b)
      for i in range(1,n):
            k=a + i*h
            if i%2 == 0:
                  result = result + 2*f(k)
            else:
                  result = result + 4*f(k)
                  result = (h/3)*result
                  print("Simpson 1/3rd Result : ", result)
                  return result
from math import *
def f(x):
      return sqrt(1+x**2)
simpsons13(f,0,10,40)
```

OUTPUT:

Simpson 1/3rd Result : 1.2644151038948792

Q.2 Write a python program to evaluate $\int_0^5 e^x dx$ by Simpson's (1/3)rd rule by considering 8 equal intervals.

Program:

```
def simpsons13(f,a,b,n):
      h = float(b-a)/n
      result = f(a) + f(b)
      for i in range(1,n):
      k = a + i*h
      if i%2 == 0:
            result = result + 2*f(k)
      else:
            result = result + 4*f(k)
            result = (h/3)*result
            print("Simpson 1/3rd Result : ", result)
            return result
from math import *
def f(x):
```

```
            return exp(x)
      simpsons13(f,0,5,8)
   OUTPUT:
      Simpson 1/3rd Result : 32.684613110896976
```

Q.3 Using Simpson's (3/8)th rule evaluate $\int_0^{10}\sqrt{1+x^2}dx$ with h= 0.2.

Program:

```
      def simpsons38(f,a,b,n):
            h = float(b-a)/n
            result = f(a)+f(b)
            for i in range(1,n):
                  k = a + i*h
                  if i%3 == 0:
                        result = result + 2*f(k)
                  else:
                        result = result + 3*f(k)
                        result = (3*h/8)*result
                        print("Simpson 3/8th Result : ", result)
                        return result
      from math import *
      def f(x):
            return sqrt(1+x**2)
      simpsons38(f,0,10,5)
```

OUTPUT:

```
     Simpson 3/8th Result : 13.318559665215194
```

Q.4 Write a python program to estimate the value of the integral $\int_0^2(x^2+2x-8)dx$ using Simpson's(1/3)rd rule, take h=0.25

Program:

```
      def simpsons13(f,a,b,n):
            h = float(b-a)/n
            result = f(a) + f(b)
            for i in range(1, n):
                  k = a + i*h
                  if i%2 == 0:
                        result = result + 2*f(k)
                  else:
                        result = result + 4*f(k)
                  result = (h/3)*result
                  print("Simpson 1/3rd Result : ", result)
                  return result
      from math import *
      def f(x):
            return x**2+2*x-8
      simpsons13(f,0,2,8)
```

OUTPUT:

Simpson 1/3rd Result : -3.145833333333333

Q.5 Write a python program to estimate the value of the integral $\int_0^5 \cos(x)\,dx$ using Simpson's(3/8)th rule, take h=0.5.

Program:

```
def simpsons38(f,a,b,n):
      h = float(b-a)/n
      result = f(a) + f(b)
      for i in range(1, n):
      k = a + i*h
      if i%3 == 0:
            result = result + 2*f(k)
      else:
            result = result + 3*f(k)
            result = (3*h/8)*result
            print("Simpson 3/8th Result : ", result)
            return result
from math import*
def f(x):
      return cos(x)
simpsons38(f,0,5,10)
```

OUTPUT:

Simpson 3/8th Result : 0.7343268508376896