_J Slip 1 L

Q.1]

2.write python code to repeat the following string 9 times using the string operator'*'.

a)python

```
string1="python"
»>   print(string1*9)
pythonpythonpythonpythonpythonpythonpythonpythonpython
```

b)mathematics

```
string="mathematics"
>»   print(string*9)
mathematicsmathematicsmathematicsmathematicsmathematicsmathematicsmathematicsmat
hematicsmathematics
```

3.write python program to generate the square of numbers from 1 to 10.

```
[(n,n*n) for n in range(1,11)]
[(1,1), (2,4), {3, 9), (4,16),(5, 25), (6, 36), (7, 49), (8, 64), {9, 81), (10, 100)]
```

Q.2]

1.using python code construct the following matrices.

a)an identity matrix of order 10*10.

```
import numpy as np
>»   print(np.eye(10))
u,.o.0.0.0.0.0.0.0.0.0J
 [0.1.0.0.0.0.0.0.0.Q0J
 [0.0.1.0.0.0.0.0.0.0J
 [0.0.0.1.0.0.0.0.0.0J
 [0.0.0.0. 1.0.0.0.Q0J
 [0.0.0.0.0.1.0.0.Q0J
 [0.0.0.0. 0.0. 1.0. 0.0J
 [0. 0.0.0. 0. 0. 0. 1. 0.0J
 [0. 0.0.0. 0.0.0.0. 1.0J
 [0.0.0. 0. 0. 0. 0. 0. 0. 1.1)
```

b) zero matrix of order 7*3.

```
>»   from sympy import*
>»   zeros(7,3)
Matrix([
[0, 0, 0),
[0, 0, 0),
[0, 0, 0),
[0, 0, 0),
[0, 0, 0],
[0, 0, 0),
[0, 0, 011)
```

c) ones matrix of order 5*4.

```
»>   from sympy import*
»>   ones{5,4)
Matrix([
[1, 1, 1, 1),
[1, 1, 1, 1),
[1, 1, 1, 1),
[1, 1, 1, 1),
[1, 1, 1, 111)
```

3.Generate all the prime numbers between 1 to 100 using python code.

7

```
>»   import math
»>   def phi(n):
          for x in range(l,100):
                  if math.gcd(100, x)==1:
                          print(x)
...
>»   phi(100)
1
3
7
9
11
13
17
19
21
23
27
29
31
33
37
39
41
43
47
49
51
53
57
59
61
63
67
69
71
73
77
79
81
83
87
89
91
93
97
99
```

Q.3]

A].1 write python program to estimate the value of the integral itegration 0 to pi sin(x)dx using simpsons (1/3)rd rule(n=6).

```
»>   def simpson13(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
            return result

>>>
»>   def f(x):
            return sin(x)

»>   from math import*
»>   simpson13(f,0,pi,6)
2.0008631896735363
```

Q3.b1)write python program to obtained the approximate real root of x"3-4x-9=0 by using regula falsi method.

```
#Program starts
#importing math module
import math

#Defining function
def f(x):
    return x**3 - 4*x - 9

# Initial values assumed
x0 = 0
x1 = 1

# Maximum number of iterations
maxiter = 10

# Error tolerance
es=  0.0001

# Iteration counter
iter = 0

# True or false value
found = False

while(found == False and iter < maxiter):
    x2 = (x0*f(x1) - x1*f(x0))/  (f(x1) - f(x0))

    if(abs(f(x2)) < es):
        found= True
    else:
        if(f(x2) * f(x1) < 0):
            x0 = x2
        else:
            x1 = x2

    iter += 1

if(found == True):
```

_J L

```
print("The approximate real root of x113-4x-9=0
```

b2).write python program to estimate the value of the integral 2 to 10 into 1/(1+x) dx using trapezoidal rule (n=B).

```
#import library
import numpy as np

#define the function
def f(x):
   return 1/(1+x)

#define the interval
a=2
b=10

#define the number of partitions
n=B

#calculate the width of the partition
h=(b-a)/n

#calculate the sum of the areas of the trapezoids
s=0.5*(f(a)+f(b))

#calculate the areas of the remaining trapezoids
for i in range (1,n):
   s=s+f(a+i*h)

#calculate the area of the trapezoidal region
l=h*s

#print the result
print("The area of the trapezoidal region is",I)
```

```
output The area of the trapezoidal region is 1.307756132756133
```

# Slip 2

_J

```
>»   def f(x):
          return x"3+5*x
... n=10
```

Q.1

1. write python code calculate the volume of a sphere with radius=7(V =4/3pir"3).

```
.»>   pi=3.14
>>> r=7.0
»>   v=4.0/3.0*pi*r**3
»>   print('the volume of the sphere is:  ',v)
the volume of the sphere is :    1436.0266666666666
>>>
```

2. Use python code to construct string operation '+' belo string.

a.string1 = Hello, string2 = World!

```
»>   string1 = "Hello"
»>   string2 = "world"
»>   string_combined= string1+string2
>»   print(string_combined)
Helloworld
>>>
```

b.string1 = Good,string2 = Morning

```
>»   string1 = "Good"
>>>  string2 = "Morning"
»>   string_combined= string1+string2
»>   print(string_combined)
GoodMorning
>>>
```

3. write python code to generate the square of numbers from 20 to 30.

```
>»   for i in [20,21,22,23,24,25,26,27,28,29,30]:
          print(i*i)

400
441
484
529
576
625
676
729
784
841
900
>>>
```

Q.2:

1.use python code find value of  f(-2),f(0),f(2) where f(x) =x"2-5x+6.

```
»>   def f(x):
          return (-2)"2-5(-2)+6
... 20


»>   def f(x):
            return (0)"2-5(0)+6
```

_J L

```
...6



»>  def f(x):
        return (2)"2-5(2)+6
...0
```

2. write python program to find the 10 term the of the sequence of function f(x)=x"3+5x.

```
»>  def f(x):
        return x"3+5*x
»>   #10th term of the sequence
... n=10
»>   print("the 10th term of the sequence is",f(n))
```

3. using sympy module of python find the eigenvalues and corresponding eigenvectors of the matrix A =([4,2,2],[2,4,2],[2,2,41).

```
»>  from sympy import*
»>  A = Matrix([[4,2,2],[2,4,2],[2,2,4]])
»>  A.eigenvalsO
{8: 1, 2: 2}
>>>

»>  from sympy import*
>>> A = Matrix([[4,2,2].[2,4,2],[2,2,4]])
»>  A.eigenvectsO
[(2, 2, [Matrix([
[-1],
[ 1],
[ 011), Matrix([
[-1],
[ O],
[ 1]])]), (8, 1, [Matrix([
[1],
[1],
[111)1)]
>>>
```

Q.3:

a.

1.write python program to estimate the value of intigral 0to1 (1/1+x"2)dx using simpson's 1/3"rd rule n=4.

```
»>  def sipmsons13(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)
```

_J

```
            result *= h/3
            return result

>>> def f(x):
            return 1/(1 +x112)
```

b.

1.write python program to obtained the aproximate real roots of x113-2x-5=0 in [2,3) using Regula_falsi method.

```
>>> def falsePosition(f,xo,x1,e):
             x0 = float(x0)
             x1 = float(x1)
             e = float(e)
             if  f(x0) * f(x1) > 0.0:
                  print('give guess values do not bracket the root')
                  print('try again with different guess values.')
             else:
                  step= 1
                  condition = true
                  while condition:
                         x2 = x0 - (x1-x0) * f(x0)/( f(x1) - f(x0) )
                         print('Iteration %d,x2 = %0.6f and f(x2) = %0.6' % (step,x2 f(x2)))
                           if f(x0) * f(x1) < 0:
                                  x1 = x2
                           else:
                                  x0 = x2
                                  step = step + 1
                 condition = abs(f(x2)) >e
                 print('\nRequired root is: %0.Bf % x2)

>>>def f(x)
            return x**3-3*x+1
```

# Slip 3

_J L

01) 1] repeat the following string 11 times using the string oprator'*' on on python
a]LATEX
B] MATLAB

```
»> string = "LATEX"
>» print(string*11)
LATEXLATEXLATEXLATEXLATEXLATEXLATEXLATEXLATEXLATEXLATEX
»>string= "MATLAB"
»> print(string*11)
MATLABMATLABMATLABMATLABMATLABMATLABMATLABMATLABMATLABMATLABMATLA
B
>>>
```

2] write a python to test whether given number is divisible by 2 or3 or 5

```
    number= int(input("Enter a number: "))

if number% 2 == 0 and number% 3 == 0 and number% 5 == 0:
    print("Number is divisible by 2, 3, and 5")
 elif number% 2 == 0 and number % 3 == 0:
     print("Number is divisible by 2 and 3")
 elif number % 2 == 0 and number % 5 == 0:
     print("Number is divisible by 2 and 5")
 elif number% 3 == 0 and number% 5 == 0:
     print("Number is divisible by 3 and 5")
elif number % 2 == 0:
    print("Number is divisible by 2")
elif number % 3 == 0:
    print("Number is divisible by 3")
elif number% 5 == 0:
    print("Number is divisible by 5")
else:
    print("Number is not divisible by 2, 3, or 5")
    output:Enter a number: 4
Number is divisible by 2
```

02

0.2.1-Using Python fing the eigenvalues and corresponding eigenvectors of the matrix([[3,-2],[6,-4]])

```
>» from sympy import*
A>» A=Matrix([[3,-2],[6,-4]])
»> A.eigenvalsO
{-1: 1, 0: 1}
>» A.eigenvectsQ
[(-1, 1, [Matrix([
[1/2],
[  1ll)l), (0,1, [Matrix([
[2/3],
[  1]])])]
```

02)2.using python code construct the following matrices.
a) an identity matrix of order 10*10.
b) zero matrix of order 7*3
c)ones matrix of order 5*4

```
import numpy as np
>» print(np.eye(10))
u1.o. 0.0. 0.0.0. 0.0.0J
 [O.1.0.0. 0.0. 0.0.0.0J
 [0.0.1.0.0.0.0.0.0.0J
 [0.0.0.1.0.0.0.0.0.0J
 [0.0.0.0.1.0.0.0.0.0J
 [0.0.0.0.0.1.0.0.0.0J
 [0.0.0.0.0.0.1.0.0.0J
 [0.0.0.0.0.0.0.1.0.0J
 [0.0.0.0.0.0.0.0.1.0J
 [O. 0. 0. 0. 0. 0. 0. 0. 0. 1.))
>>> from sympy import*
>>> zeros(7,3)
Matrix([
[O, 0, 0),
[O, 0, 0),
[O, 0, 0),
[O, 0, 0),
[O, 0, O].
[O, 0, 0),
[O, 0, O]])
>>> from sympy import*
>>> ones(5,4)
Matrix([
[1, 1, 1, 1),
[1, 1, 1, 1),
[1, 1, 1, 1),
[1, 1, 1, 1),
[1, 1, 1, 1)))
```

Q3.A1).write a python program to estimate the value of the integral [pi sin (x)dx using simpson's(1/3)rd result(n=6)]

```
>>>def simpson13(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
                    return result

 >>>
 >>>def   f(x):
                return sin(x):

 >>> from math import *
 >>>simpson13(f, 0, pi, 6)
 6.002589569020609
```

Q3. b2.) write python program to estimate the value of the integral 2 to 10 into 1/(1+x)dx using trapezoidal rule (n=5).

```
#importing math library for calculation
import math
```

_J

```
#defining the function
def f(x):
    return 1/(1 + x)

#defining the interval
a=2
b=10

#number of trapezoids
n=5

#calculating h
h = (b - a)/ n

#calculating sum of first and last terms
s = f(a) + f(b)

#calculating sum of remaining terms
for i in range(1, n):
    s = s + 2 * f(a + i * h)

#calculating value of integral
integral = (h/2) * s

#printing result
print("The value of integral is: ",integral)
OUTPUT: The value of integral is:  1.3206255135651455
```

# Slip 4

J L

Q-1)Attempt any two of the following.
1)using python code sort the tuple in ascending and descending order 5,-3,0,1,6,-6,2.

Ascending

```
aTuple = (5,-3,0,1,6,-6,2)
>»  result= sorted(aTuple)
»>  result = Tuple(result)
»>  result = tuple(result)
>»  print('Sorted Tuple:',result)
Sorted Tuple: (-6, -3, 0, 1, 2, 5, 6)
```

Descendig

```
aTuple = (5,-3,0,1,-6,2)
»>result= sorted(aTuple,revesre=True)
»>  result = tuple(result)
»>  print('Sorted Tuple:',result)
Sorted Tuple: (5, 2,1, 0, -3, -6)
```

2)write python program which deals with concatenataion repetition of lists.
list1 = [15,20,25,30,35,40]
list2 = [7,14,21,28,35,42]

```
list1 = [15,20,25,30,35,40]
>»  list2 = [7,14,21,28,35,42]



»>  res = list1 + list2
»>  print ("concatenated list:\n"+ str(res))
concatenated list:
[15, 20, 25, 30, 35, 40, 7, 14, 21, 28, 35, 42]
>>>
```

a) Find List1 + List2

```
>»  list1+list2
[15, 20, 25, 30, 35, 40, 7, 14, 21, 28, 35, 42]
```

b) Find 9*List1

```
>»  list1*9
[15,20,25,30,35,40, 15,20,25,30,35,40,15,20,25,30,35,40,15,20,25,30,35,40,15,20,25,
30,35,40, 15,20,25,30,35,40,15,20,25,30,35,40,15,20,25,30,35,40,15,20,25,30,35,40]
```

c) Find 7*List2

```
»>  list2*7
[7,14,21,28,35,42, 7,14,21,28,35,42, 7,14,21,28,35,42, 7,14,21,28,35,42, 7,14,21,28,35,
42, 7, 14,21,28,35,42,7,14,21,28,35,42]
```

3) write python code to find the square of odd numbers from 1 to 20 using while loop.

```
>»  for i in [1,3,5,7,9,11,13,15,17,19]:
        print(i*i)

1
9
25
49
81
```

_J

```
121
169
225
289
361
>>>
```

02

1. using python code construct the following matrices.

a) an identity matrix of order 10*10.

```
import numpy as np
>>> print(np.eye(10))
[[1. 0. 0. 0. 0. 0. 0. 0. 0. O.]
 [O.1.0.0.0.0.0.0.0.0J
 [0.0. 1.0.0.0.0.0.0.0J
 [0.0.0.1.0.0.0.0.0.0J
 [0.0.0.0.1.0.0.0.0.0J
 [0.0.0.0.0. 1.0.0.0.0J
 [0.0.0.0.0.0. 1.0.0.0J
 [0.0.0.0.0.0.0.1.0.0J
 [0.0.0.0.0.0.0.0. 1.0J
 [0.0.0.0.0.0.0.0.0. 1J
```

b) zero matrix of order 7*3

```
»>  from sympy import*
»>  zeros(7,3)
Matrix([
[0, 0, 0],
[0, 0, 0],
[0, 0, 0],
[0, 0, 0],
[0, 0, 0],
[0, 0, 0],
[O, 0, 011)
```

c) Ones matrix of order 5*4

```
»>  from sympy import*
>>> ones(5,4)
Matrix([
[1, 1, 1, 1],
[1, 1, 1, 1),
[1, 1, 1, 1),
[1, 1, 1, 1),
[1, 1, 1, 111)
```

2)Find the type of the following data by using python code

```
  type('number')
<class 'str'>
>>> type(31.25)
<class 'float'>
```

```
>>> type('Mathematics')
<class 'str'>
>>> type(49)
<class 'int'>
>>>
```

Q.3
1)write python program to estimate the value of the integral pi into 0 xsin(x)dx using simpson's(1/3)rd rule (n=6)

```
>>> def simpson13(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
        return result

>>> def f(x):
        return sin(x)

>>> from math import *
>>> simpson13(f,0,pi,6)
 002589569020609
>>>
```

Q3-b1) write python program to find all positive prime numbers less then given number n.

```
def counLprimes_nums(n):
        ctr= 0
        fornum in range(n):
            if num <= 1:
                continue
            for i in range(2,num):
                if (num % i) ==0:
                    break
            else:
                ctr +=1
        return ctr
    print(count_primes_nums(10))
>>> print(count_primes_nums(10))
None
>>> print(count_primes_nums(100))
None
```

_J

Q1)attempt any two the following

2)Evaluate following expression on python

a)m=[1,2,3,4],find length

Slip 5

```
import math
»>  m=[1,2,3,4]
»>  print(m)
[1, 2, 3, 4]
```

b) l='XYZ'+'pqr',find L

```
  import math
»>  l='XYZ'
»>  l='pqr'
>»  print(l)
pqr
```

c) s='make in india',find(s[:71)&(s[:91)

```
»  import math
»>  s='make in india'
»>  s(s[:7[&(s[:91)
»>  print(s)
make in india
```

3)use python code to genrate the square root numbers from 21to49

```
»>  import math
»>  print("square root of 21 is:",math.sqrt(21))
square root of 21 is: 4.58257569495584
»>  import math
»>  print("square root of 49 is:",math.sqrt(49))
square root of 49 is: 7.0
```

q2)

1)using python construct the following matricx 1)an matrix of order 1OX10

```
import numpy as np
x = np.ones((10, 10))
x[1:-1, 1:-1] = 0
print(x)
output::
I[ 1.   1.   1.   1.   1.  1.   1.   1.  1.   1.]
  [ 1.   0.   0.   0.   0.  0.   0.   0.   0.   1.]
  [ 1.   0.   0.   0.   0.  0.   0.   0.   0.   1.]
  [ 1.   0.   0.   0.   0.  0.   0.   0.   0.   1.]
  [ 1.   0.   0.   0.   0.  0.   0.   0.   0.   1.]
  [ 1.   0.   0.   0.   0.  0.   0.   0.   0.   1.]
  [ 1.   0.   0.   0.   0.  0.   0.   0.   0.   1.]
  [ 1.   0.   0.   0.   0.  0.   0.   0.   0.   1.]
  [ 1.   0.   0.   0.   0.  0.   0.   0.   0.   1.]
  [ 1.   1.   1.   1.   1.  1.   1.   1.   1.   1.1]
```

2) zero matricx of order 7X3

```
>»zeros (7,3)
matrix([
[0,0,0],
[0,0,0],
[0,0,0],
[0,0,0],
```

7

```
[0,0,0],
[0,0,0),
[0,0,0)))
```

3)ones matrix of order 5*4

```
>>>from sympy import*
>>>ones(5,4)
matrix([
[1,1,1,1),
[1,1,1,1),
(1,1,1,1),
(1,1,1,1),
[1,1,1,111)
```

2)using linsolve command in python solve the follwing system of linear equation

x-2y+3z=7

2x+y+z=4

-3x+2y-2z=-10

```
>>>from sympy import
>>>x,y,z= symbols("x,y,z")
>>> A=matrix([[l,2,3),[2,1,1),(-3,2,-211)
>>>b=matrics([7,-4,-101)
>>>linsolve((A,b),[x,y,z])
finteset((z-1,2-2*z,z))
```

Q3)

a)

1)write python code to find eignvalue and corresponding eigenvector of the matrix and hence find matrix p with diagonalize to A

```
>>>from sympy import*
>>>A=matrix([1,3,3],[2,2,3],[4,2,11)
>>>A.diagonlize()
```

b)write python program to cvalute f(3.5)by differnce formula of the given data

| x | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| y=f(x) | 41 | 62 | 65 | 50 | 17 |

```
#import data
x=np.array([l ,2,3,4,51)
y=np.array([41,62,65,50,171)
#calulate fourth order toward differnce
arr=np.zeros(len(x)-4)
for i in range(4,len(x));
    arr[i-4]=(-y[i)+4*y[i-1]-6*y[i-2)+4*y[i-3]-y[i-41)/(x[i])
 #print the output
 print("fourth order toward differnce:")
 print(arr)
```

J L

slip no.6
Q-1

2. write a python program to list name and roll no. of 5 students in BSC(computer scirnce)

```
     students= [
        {'name': 'John', 'roll_number':1},
        {'name': 'Steve', 'roll_number': 2},
        {'name': 'Karen', 'roll_number': 3},
        {'name': 'Sophia', 'roll_number': 4},
        {'name': 'Alice', 'roll_number': 5}
... ]
»>   for student in students:
        print(f"Name: {student['name']}, Roll number: {student['roll_number']}")

Name: John, Roll number: 1
Name: Steve, Roll number: 2
Name: Karen, Roll number: 3
Name: Sophia, Roll number: 4
Name: Alice, Roll number: 5
```

3. write a python program to find maximum and minimum elements in the given list
[7, 8, 71, 32, 49, -5, 7, 7, 0, 1, 6]

```
»>   listl = (7, 8, 71, 32, 49, -5, 7, 7, 1, 6]
»>   print("Largest element is:",max(listl))
Largest element is: 71
»>   print("Smallest element is:",min(list1))
Smallest element is: -5
```

Q-2
1. using python code construct identity matrix of order 10 and hence find determinant.trace and transpose of it.

```
»>   import numpy as np
»>   ldentity_matrix = np.identity(10)
»>   print("Identity Matrix of order 10 is:  \n", ldentity_matrix)
Identity Matrix of order 10 is :
 rr1. o.o.o.o.o.o.o.o.oJ
 [0.1. 0.0.0.0.0.0.0.0.0J
 [0.0. 1.0.0.0.0.0.0.0.0J
 [0.0. 0.1.0.0.0.0.0.0.0J
 [0.0.0.0. 1.0.0.0.0.0.0J
 [0.0.0.0.0. 1.0.0.0.0.0J
 [0.0. 0.0.0.0. 1.0.0.0.0J
 [o.o. 0.0.0.o.o.1.o.oJ
 [0.0. 0.0.0.0.0.0.1.0J
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]]
>>>
»>   # Determinant of an identity matrix is always 1
»>   determinant = np.linalg.det(ldentity_matrix)
»>   print("\nDeterminant of the matrix is:", determinant)

Determinant of the matrix is: 1.0
>>>
»>   # Trace of an identity matrix is always equal to the order of the matrix
```

7

```
>>>trace= np.trace(Identity_matrix)
>>> print("\nTrace of the matrix is: ",trace)

Trace of the matrix is:    10.0
>>> >>>
>>> # Trace of an identity matrix is always equal to the order of the matrix
>>> trace = np.trace(Identity_matrix)
>>> print("\nTrace of the matrix is: ", trace)

Trace of the matrix is:    10.0
>>>
>>> # Transpose of an identity matrix is same as it is
>>>transpose= np.transpose(Identity_matrix)
>>> print("\nTranspose of the matrix
```

2.write a python code to find the value of function f(x,y)=x...2-2xy+4 at the points(2,0) (1,-1)

```
»>  def f(x,y):
      return x**2 - 2*x*y + 4

>>> # point (2,0)
>>> print(f(2,0))
8
>>>
»>  # point (1,-1)
»>  print(f(1,-1))
7
```

3. write number between 1 to 200 which are divisible by 7 using python code

```
>>>list=☐
>>> for x in range(1,201):
        if x % 7 == 0:
             list.append(x)

»>  print(list)
[7,14,21,28,35,42,49,56,63, 70, 77,84,91,98, 105,112,119,126,133,140,147,154,161,168,
175, 182, 189, 196)
>>>
```

Q-3

A) 1.write a python program to diagonalize matrix
A([[3,-2],[6,-411)
and find the matrix P with diagonalize of A and diagonal matrix D

```
»>  import numpy as np
>>>
»>  # Define the matrix
»>  A = np.array([[3,-2],[6,-411)
>>>
»>  # Calculate eigenvalues and eigenvectors
»>  eigen_values, eigen_vectors = np.linalg.eig(A)
»>  >»  # Define the diagonal matrix
```

```
>>> D = np.diag(eigen_values)
>>>
>>> # Calculate the matrix P
>>> P  = np.matmul(eigen_vectors, np.linalg.inv(D))
>>>
>>> print("The matrix P is:\n", P)
The matrix P is:
 [[    3.12268450e+14-4.47213595e-01]
 [  4.68402674e+14 -8.94427191e-01]]
>>> print("The diagonal matrix D is:\n", D)
The diagonal matrix D is:
 [[  1.77635684e-15    0.00000000e+00]
 [ 0.00000000e+00 -1.00000000e+00]]
>>>
```

Q-3

A) 1.write a python program to diagonalize matrix
A([[3,-2],[6,-4]])
and find the matrix P with diagonalize of A and diagonal matrix D

```
>>> import numpy as np
>>>
>>> # Define the matrix
>>>A= np.array([[3,-2],[6,-4]])
>>>
>>> # Calculate eigenvalues and eigenvectors
>>> eigen_values, eigen_vectors = np.linalg.eig(A)
>>> >>> # Define the diagonal matrix
>>> D = np.diag(eigen_values)
>>>
>>> # Calculate the matrix P
>>> P  = np.matmul(eigen_vectors, np.linalg.inv(D))
>>>
>>> print("The matrix P is:\n", P)
The matrix P is:
 [[    3.12268450e+14-4.47213595e-01]
 [  4.68402674e+14 -8.94427191e-01]]
>>> print("The diagonal matrix D is:\n", D)
The diagonal matrix D is:
 [[  1.77635684e-15    0.00000000e+00]
 [ 0.00000000e+00 -1.00000000e+00]]
>>>
                x3 = (x1 * f_x2 - x2 * f_x1) / (f_x2 - f_x1)
                # calculating value of function at
```

# Slip 8

_J l

q1] 1).use python code to find a+c,ab,c"d,a/b and a{b+c),whwre a=5, b=7, c=9, d=11.

```
a+c = 14
ab=  35
c"d = 3486784401
a/b = 0.7142857142857143
a(b+c) = 160
```

2).the following two statements using the'+' string operation on python.
a. string1 = india won, string2 = world cup
b. string1 = god, string2 = is great

c. 'India won the World Cup!'
d. 'God is great!'

3).write python code to find area and circumference of circle with radius 14.

```
# Area of circle
area= 3.14 * {14    2)

# Circumference of circle
circumference = 2 * 3.14 * 14

print{"Area of circle = ", area)
print{"Circumference of circle = ·, circumference)

output:Area of circle=    615.44
Circumference of circle=    87.92
```

q2].1).using python code logically verify associativity of matrices with respective to matrix addition (use proper matrices).

```
#importing numpy
import numpy as np

#creating 3 matrices
A = np.array{[[2,3], [1,3]])
B = np.array{[[1,2], [3,411)
C = np.array{[[3,3], [2,1ll)

#step 1: verifying A+{B+C)==(A+B)+C
result1 = A + (B + C)
result2 = (A+  B) + C

#step 2: checking if the result1 and result2 are equal
if np.array_equal{result1, result2):
   print{"Matrix Addition is associative")
else:
   print{"Matrix Addition is not associative")

output: Matrix Addition is associative
```

_J L

2).write python code to generate 10 terms of fibonacci sequence using loop.

```
#initializing the first 2 terms
a=O
b=1

#using loop to generate and print 10 terms
for i in range(10):
    print(a,end=" ")
    c=a+b
    a=b
    b=c
output O1 1 2 3 5 8 13 21 34
```

q3]. a1).write python program to estimate the value of the integral Oto 1 into 1/(1+xA2)into dx using simpsons (1/3)rd rule (n=6).

```
import numpy as np

def f(x):
    return 1/(1 + xtt2)

# Function to calculate area
def simpson(a, b, n):

    # Calculating h
    h = (b - a)/n

    # Calculating result
    result= 0
    for i in range(O, n + 1):
        if i == 0 or i == n:
            result+= f(a + i * h)
        elif i % 2 != 0:
            result+= 4 * f(a + i * h)
        else:
            result+= 2 * f(a + i * h)

    result = result * (h / 3)
    return result

# Driver code
a=O
b=1
n=6

print(simpson(a, b, n))

output 0.7853979452340107
```

q3].a2). write python program to evaluate fourth order forward difference of the given data.

**x 1 2 3 4 5**
y=f(x) 41 62 65 50 17

```
#import numpy
import numpy as np

#Input data
x = np.array([l ,2,3,4,5])
y = np.array([41,62,65,50,17))

#Calculate fourth order forward difference
arr = np.zeros(len(x)-4)
for i in range(4,len(x)):
        arr[i-4] = (-y[i] + 4")t[i-1] - 6*y[i-2] + 4*y[i-3] - y[i-4))/(x[i] - x[i-4])

#Print the output
print("Fourth order forward difference:")
print(arr)
```

q3.b1). write pyton program to obtained the approximate real root of x"3-2x-5=0 in [2,3] using regula-falsi method

```
def f(x):
        return x**3-2*x-5

# take two points
a,b=2,3

# Iterating till the root is found
for i in range(50):
        c= (a*f(b)-b*f(a))/(f(b)-f(a)) #regula falsi formula
        if f(c)==0:
                break
        elif f(c)*f(a) < 0:
                b=c
        else:
                a=c

print("The root is",c)
```

```
output The root is 2.0945514815423265
```

q3.b2).write python program to estimate the value of the integral 2 to 4 into (2x"2-4x+1) dx using trapezoidal rule (n=5).

```
import numpy as np

# define the function
def f(x):
        return (2*x**2 - 4*x + 1)
```

```
# Trapezoidal rule
# n is the number of trapezoids
def trapezoidal(a, b, n):
    # Grid spacing
    h = (b - a)/  n

    # Computing sum of first and last terms
    # in above formula
    s = f(a) + f(b)

    # Adding the terms in between
    # the first and last terms
    for i in range(1, n):
        s += 2 * f(a + i * h)

    # h/2 indicates (b-a)/2n.
    # Multiplying h/2 with s
    return (h / 2) * s

# Driver code
a = 2 # lower limit of integration
b = 4 # upper limit of integration
n = 5 # no. of trapezoids

# printing the value of
```

Q.1:1)

using python evaluate each of the following expression
a) 30 modulus 2+7 -(3+9)*20/5
b) 30*10 floor division 3+30 modulus 3
c) 5"5-5"3+7 floor division 7

Slip 9

Answers:-

```
def _init_(self, path):
          dirname = os.path.dirname(path)
          os.makedirs(dirname, exist_ok=True)
          f = open(path, "a+")

          # Check that the file is newline-terminated
          size = os.path.getsize(path)
          if size> 0:
               f.seek(size - 1)
               end = f.read(1)
               if end != "\n":
                    f.write("\n")
          self.f = f
          self.path= path

     def log(self, event):
          event["_event_id"] = str(uuid.uuid4())
          json.dump(event, self.f)
          self.f.write("\n")

     def state(self):
          state = {"complete": set(), "last": None}
          for line in open(self.path):
               event = json.loads(line)
               if event["type"] == "submit" and event["success"]:
                    state["complete"].add(event["id"])
                    state["last"] = event
          return state
```

Q.1:2)

use print command on python to find
a) sin30
b)pi
c) e
d) cos30

Answers:-
a) print(math.sin(30))
output
-0.9880316240928618
b) print(math.pi)
output
3.141592653589793
c) print(math.e)
output
2.718281828459045
d) print(math.cos(30))

output
0.15425144988758405

Q.1:3)

write python code to generate modulus value of -10,10,-1,1,0.

Answers:-

```
mod = 0
list = [-10,10,-1,1, OJ
for i in list
      mod= abs(i)
      print(mod)

# Output
#10
#10
#1
#1
#0
```

Q.2:-1}
use python code to generate second,fifth,eight characters from string 'MATHEMATICS'

Answers:-
print('MATHEMATICS'[1], **'MATHEMATICS'[4], 'MATHEMATICS'[?]}**

Q.2:2

using python find the eigenvalues and corresponding eigenvectors of the matrix [[3 -2][6 -4]]

Answers:-

```
import numpy as np
A= np.array([[3, -2), [6,-4]])

w, v = np.linalg.eig(A}

print("Eigenvalues:", w)

print("Eigenvectors:", v)
```

output-

```
1 import numpy as np
      2 A = np.array([[3, -2), [6,-4)))
      3
      4 w, v = np.linalg.eig(A)
```

Q.2:3

write python code to verify (AB}"-1=B"-1A"-1(use proper matrices A and B)

Answers:

```
import numpy as np
```

output-

```
ABA = np.array([[2, 3], [4, 511)
B  =  np.array([[1, 5]. [3, 711)

A_inv = np.linalg.inv(A)
B_inv = np.linalg.inv(B)_inv = np.dot(B_inv, A_inv)

print(AB_inv)
```

Q.3:1

write python program to estimatethe value of the integral 1 negation 1O(x"2+5x)dx using simpson's(1/3)"rd rule (n=5)

Answers:-

```
#Python  Program
import numpy as np

#Simpson's(1/3)rd Rule
def simpsons_1_3rd_rule(f,a,b,n):
   h=(b-a)/n   x=np.linspace(a,b,n+1)
   fx=f(x)
   s=fx[O]+fx[n]
   for i in range(1,n):
      if i%2==1:
         s+=4*fx[i]
      else:
         s+=2*fx[i]
   return (h/3)*s

#defining the function
def f(x):
   return x""""2+5*x

#defining the lower and upper limit
output:-
a=1
b=10

#defining the number of intervals
n=5

#calling the simpsons's(1/3)rd rule
I=simpsons_1_3rd_rule(f,a,b,n)

#printing the value of the integral
print("The value of the integral is",I
```

Q.3:2

write python program to evaluate interpolate value f( 2.5)of the given data

|  |  |  |  |  |
|---|---|---|---|---|
| X | 1 , | 2 , | 3 , | 4 |
| y=f(x) | 1 , | 8 , | 27, | 64 |

Answers:

```
# Program
import numpy as np

# Input data
output:-
x = np.array([1, 2, 3, 41)
y = np.array([1, 8, 27, 64))

# Interpolated value at x = 2.5
x_interpolated = 2.5

# Linear Interpolation
y_interpolated = np.interp(x_interpolated, x, y)

print("The interpolated value at x=2.5 is", y_interpolated)
```

Q.3:B:1

write python program to obtained the approximate real root of xA3 - 4x - 0 by using regula-falsi method.

Answers

```
def regulaFalsi(f, x0, x1, e):
    x2 = 0
    while(True):
        x2 = (x0 * f(x1) - x1 * f(x0))/ (f(x1) - f(x0))
        if abs(f(x2)) < e:
            break
        # Check if x2 is root of
        # equation or not
        if f(x0) * f(x2) < 0:
            x1 = x2
        else:
            x0 = x2
    return x2

# Function to find the root
def func(x):
    return x * x * x - 4 * x

# Driver Code
x0 = 0
x1 = 1
e = 0.0001

 print("The root of the given equation is :",
                 regulaFalsi(func, x0, x1, e))
```

Q.3:b:2
write python program to evaluate fourth order forward difference of the given data

| **x** | **1,** | **2** | **,3** | **4,5** |
|---|---|---|---|---|
| y =f(x) | 40 | 60 , | 65 , | 50, 18 |

Answers:-

```
# Python Program to evaluate 4th order forward difference of the given data

# Define x and y
output:-
x = [1, 2, 3, 4, 5]
y = [40, 60, 65, 50, 18]

# Calculate the 4th order forward difference
diff4 = □
for i in range(4):
    diff4.append{y[i + 4] - 4 * y[i + 3] + 6 * y[i + 2] - 4 * y[i + 1] + y[i])

# Print the 4th order forward difference
print{"4th Order Forward Difference is :", diff4)
```

_J

Q1:
2. Using Python code
List1 = (5, 10, 15, 20, 25, 30] and List2 = [7, 14, 21, 28, 35, 42]
Evaluate
(a) List1 + List2
(b) 3*List1
(c) 5*List2

Slip 10

Answer:-

list1=[5,10,15,20,25,30]
list2=[7,14,21,28,35,42]

A) list1+list2
OUTPUT
[5, 10, 15, 20, 25, 30, 7, 14, 21, 28, 35, 42]

B) 3*1ist1
OUTPUT
[15, 20, 25, 30, 35, 40]

C) 5*1ist2
OUTPUT
(35, 70, 105, 140, 175, 21O]

Q.1:
3. Write Python code to find area of triangle whose base is 10 and height is 15.

Answer:-

```
# Area of a triangle = (base * height) / 2
area = (10 * 15) / 2
print("Area of triangle with base 10 and height 15 is:", area)
```
OUTPUT
```
Area of triangle with base 10 and height 15 is: 75.0
```

Q.2:
1. using python code construct the following matrices.

a) an identity matrix of order 10*10.

b) zero matrix of order 7*3

c) ones matrix of order 5*4

Answer

```
import numpy as np
>» print(np.eye(1O))
u1.o.o.o.o.o.o.o.o.o.0J
 [0.1.0.0.0.0.0.0.0.0J
 [0.0.1.0.0.0.0.0.0.0J
 [0.0.0.1.0.0.0.0.0.0J
 [0.0.0.0. 1.0.0.0.0.0J
 [0.0.0.0.0. 1.0.0.0.0J
 [0.0.0.0.0.0. 1.0.0.0J
```

7

```
   .0.0.0.0.0.0. 1.0.0J
   .0.0.0.0.0.0.0. 1.0J
  [O. 0.  0. 0. 0. 0. 0. 0. 0. 1.]]
>>> from sympy import*
>>> zeros(7,3)
Matrix([
[0, 0, O],
[0, 0, O],
[0, 0, O],
[0, 0, O],
[0, 0, O],
[0, 0, O],
[O, 0, 011)
>>> from sympy import*
>>> ones(5,4)
Matrix([
[1, 1, 1, 1],
[1, 1, 1, 1],
[1, 1, 1, 1],
[1, 1, 1, 1],
[1, 1, 1, 111)
```

02:
2) Write Python program to find the value of function f (x) = x2 + x, (-5 s x s 5).

```
# Solution

# creating a list with the range of x
x_list = list(range(-5, 6))

# creating an empty list for storing the values of f(x)
t_x_list = ☐

# using for loop to calculate the value of f(x)
for x in x_list:
      L x = x * * 2 + x
      t_x_list.append(f_x)

# printing the list of values of f(x)
print("The list of values of f(x) is:", t_x_list)
```

OUTPUT

```
The list of values of f(x) is: [20, 12, 6, 2, 0, 0, 2, 6, 12, 20, 30]
```

Q.2
3)Write Python program to find the determinant of matrix
A=
0
0
0
1 0 5
21 6
340
0
0
0 and B =
[

```
25
-1  4
]
```

ANSWER:-

```
# importing numpy library
import numpy as np

# define matrix A
A = np.array([[1, 0, 5], [2, 1, 6], [3, 4, 011)

# define matrix B
B = np.array([[2, 5], [-1, 411)

# calculate the determinant of A
determinant_A = np.linalg.det(A)
print("Determinant of A is :", determinant_A)

# calculate the determinant of B
determinant_B = np.linalg.det(B)
print("Determinant of B is :", determinant_B)
```

OUTPUT

```
Determinant of A is:  0.9999999999999967
Determinant of B is : 13.0
```

Q.3 A)

2. Write Python program to evaluate interpolated value f (2.7) of the given data f(2)=0.69315,f(2.5)=0.91629,f(3)=1.09861.

Answer

```
import numpy as np
#Given Data
x = np.array([2,2.5,31)
y = np.array([0.69315, 0.91629, 1.098611)

#Evaluating f(2.7) using Lagrange's Interpolation formula
def lagrange_interpolation(x_val, x, y):
        f_x_val = 0
        n = len(x)
        for i in range(n):
                term= y[i]
                for j in range(n):
                        if j != i:
                                term = term*(x_val - xU])/(x[i] - xU])
                t_x_val += term
        return t_x_val

#Evaluating f(2.7)
print("f(2.7) is", lagrange_interpolation(2.7, x, y))
```

output

```
f(2.7) is 0.9941164
```

Q.3 B)

1. Write Python program to obtained the approximate real root of x3 - 4x - 9 = 0 by using Regula-falsi method.

Answer

```
# Python Program to Obtain the Real Root of x3 - 4x - 9 = 0
# Using Regula-Falsi Method

def func(x):
    return x**3 - 4*x - 9

# Prints root of the equation x3 - 4x - 9 = 0
def regulaFalsi(a, b):

    if func(a) * func(b) >= O:
        print("You have not assumed right a and b")
        return -1

    c = a # Initialize result

    for i in range(500):

        # Find the point that touches x axis
        c = (a*func(b) - b*func(a))/ (func(b) - func(a))

        # Check if the above found point is root
        if func(c) == O:
            break

        # Decide the side to repeat the steps
        elif func(c)*func(a) < O:
            b=c
```

Q.3
b)Write Python program to estimate the value of the integral *f* 1
0 cos(x)dx using Trape-
zoidal rule (n=5).
ANSWER:-

```
import math

def trapezoidal(n):
    h = 1.0 / n
    s = 0.0
    for i in range(n):
        s += (h / 2.0) * (math.cos(h * (i +1)) + math.cos(h * i))
    returns

n=5
print("Value of the integral is :", trapezoidal(n))
```

OUTPUT:-

```
Value of the integral is : 0.8386642098070081
```

_J

slip no:-11

Q.1)Attempt any two of the following.

1.evaluate the following expression in Python.

(a) M=[1,2,3,4,5,6,7],find length M.

Ans:->» # Python code to demonstrate string length

```
>>> # using len
>>> str =(1,2,3,4,5,6,7]
>>> print(len(str))
7
```

(b) L="XY"+"pqr",find L.

Ans:->>> L = **"XV"** + "pqr"

```
»>  print(L)
XYpqr
```

(c) s='Make In India',find (s[:51)&(s[:91).

Ans:->» s='Make In India'

```
»>  print(s(:51)
Make
>»  print(s(:91)
Make In I
```

2.Write Python code to reverse the string S=[3,4,5,6,7,8,9,10,11,12,13].

Ans:->>> def reverse(itr):

```
        return itr(::-1]

>»  itr1 = '3,4,5,6,7,8,9,10,11,12,13'
>»  print("Original string:",itr1)
Original string: 3,4,5,6,7,8,9,10,11,12,13
>»  print("Reverse string:",reverse('3,4,5,6,7,8,9,10,11,12,13'))
Reverse string: 31,21,11,01,9,8,7,6,5,4,3
```

Q.2)Attempt any two of the following.

1.Using Python code to list Name of 5 teacher in your college with their subject.

Ans:->>># In[]:

```
>»  #Using Python code to list Name of 5 teacher in your college with their subject
>>>
>»   teacher_name = ('Mr.om','Mr.sanket','Mr.tejus','Mr.kartik','Mr.ajay']
>»  teacher_subject = ('Maths','Physics','Chemistry','Biology','English']
>>>
>»  for i in range(len(teacher_name)):
        print(teacher_name[i],'teaches',teacher_subject[i])

Mr.om teaches Maths
Mr.sanket teaches Physics
Mr.tejus teaches Chemistry
Mr.kartik teaches Biology
Mr.ajay teaches English
```

2.Generte all the prime numbers between 51 to 100 using Python program.

Ans:->>> # Python program to display all the prime numbers within an interval

```
>»  lower = 51
>»  upper = 100
>»  print("Prime numbers between", lower, "and", upper, "are:")
    Prime numbers between 51 and 100 are:
```

7

J L

```
>>> 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)
Prime numbers between 51 and 100 are: 53, 59, 61, 67, 71, 73, 79, 83, 89,97
```

Q.3).a.Attempt any one of the following

1.Write Python program find the approximate root of the function xA5+3x+1,in[-2,0]using Newton Raphson Method correct upto 4 decimal places.

Ans:->>> import math

```
>>> def f(x):
          return x"""5+3*x+1

>>> def f1(x):
          return 5*x**4+3

>>> def newton(x):
          h=f(x)/f1(x)

>>> while abs(h)>=0.0001:
               h=f(x)/f1(x)
               x=x-h

>>> print("The value of the root is:",x)
>>> x=int(input("Enter the value of x:"))
Enter the value of x:newton(x)
```

b.Attempt any one of the following

1. Write Python program to obtained the approximate real root of xA3-4x-9=0 by using Regula-falsi method.

Ans:->>> # Implementation of Linear Interpolation using Python3 code

```
>>> # Importing library
>>> from scipy.interpolate import interp1d
>>> X = [150,1 52,154,155]#random x values
>>> Y = [12.247,12.329,12.410,12.490]#random y values
>>> #test value
>>> interpolate_x = 153
>>> #Finding the interpolation
>>> y_interp = interp1d(X,Y)
>>> print("Values of Y at x = O is".format(interpolate_x),y_interp(interpolate_x))
Values of Y at x = 153 is 12.3695
```

_J

01

q1. using python evaluate each of the following expresion.

```
>»  23 % 2+9-(3+7)*10/2
-40.0
»>  35 * 10 / 3 +15 % 3
116.66666666666667
»>  3"5 -2"5 + 4 // 7
46
```

Slip 12

q2. use while command on python to find odd positive integer between 25 to 50.

```
print([even for even in range(25,51)if even%2!=01)
[25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]
>>>
```

02

q2 write python program to find the product of n natural numbers using while loop.

```
»>  product=1
>>> n=1
»>  while n<= 10:
         product=product*n
         n=n+1
...
>»  print('product =',product)
product = 3628800
>>>
```

q3 Generate all prime numbers between 1 to 200 using python code.

```
»>  import math
»>  def phi(n):
         for x in range(1,200):
                if math.gcd(200,x)==1:
                     print(x)

»>  phi(200)
1
3
7
9
11
13
17
19
21
23
27
29
31
33
37
```

7

Q3

A)

q1 write python program to estimate the value of the integral integral integration 0 to pi sin(x)dx using simpson's (1/3rd rule (n=5).)

```
»>  def simpson13(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
         return result

»>  def f(x):
         return sin(x)

>>>
»>  from math import *
>»  simpson13(f,0,pi,5)
1.9337655980928052
>>>
```

B)

```
»>  def falseposition(f,x0,x1,e):
         x0 = float(x0)
         x1 = float(x1)
         e = float(e)
         if f{x0) * f(x1) >0.0:
              print('given guess values do not bracket the root.')
              print('try again with different quess values.')
         else:
              step= 1
              condition = true
              while condition:
                     x2 =x0 - (x1-x0) * f{x0}/(f(x1)-f(x0))
              print('iteration %d, x2 = %0.6f and f(x2)= %0.6f'%(step,x2,f(x2)))
```

7

_J L

```
              if f(x0) * f(x2)<0:
                     x1 = x2
              else:
                     x0 = x2
              step = step + 1
              condition = abs(f(x2))>e
      print('\nrequired root is: %0.8f% x2)
   File "<stdin>", line 20
     print('\nrequired root is: %0.8f% x2)
```
,.

Slip 14

Q1. Use print code on Python(a=4,b=6,c=8,d=12).

```
a=4
>>> b=6
>>> c=8
»>  d=12
»>  print(a+c)
12
»>  print(a*b)
24
»>  print(c**d)
68719476736
»>  print(a/b)
0.6666666666666666
>>>
»>  3+(9-2)/7*2**2
7.0
```

2. For the following two statements use'+'string operation on python.

a)string1=Hello,string2=World!

```
string1='Hello'
»>  string2='World'
»>    string_combined=string1+string2
»>  print(string_combined)
HelloWorld
```

b)string1=Good,string2=Morning

```
string1="Good"
»>  string2="Morning"
»>    string_combined=string1+string2
»>   print(string_combined)
GoodMorning
```

3.Use Python loop to print('Hello','i','You Learn Python')
where i=['Saurabh' ,'Akash' ,'Sandeep' ,'Ram' ,'Sai']

```
i='Saurabh'
print('Hello','Saurabh','You Learn Python')
Hello Saurabh You Learn Python
i='Akash'
»>   print('Hello','Akash','You Learn Python')
Hello Akash You Learn Python
i='Sandeep'
»>   print('Hello','Sandeep','You Learn Python')
Hello Sandeep You Learn Python
i='Ram'
»>   print('Hello','Ram','You Learn Python')
Hello Ram You Learn Python
```
**i='Sai'**
```
»>   print('Hello','Sai','You Learn Python')
Hello Sai You Learn Python
```

Q2. Attempt any two of the following.

1)Using Python code construct any two matrices A and B

1)Show that **A+B=B+A.**

```
     A=([[1,2,3],[3,4,511)
»>  8=([[1,2,3],[4,5,611)
»>  #Matrix addition
»>  A+B=B+A
```

```
A+B==B+A
False
from sympy import*
>»  A=Matrix([[4,2,2],[2,4,2]])
>»  B=Matrix([[1,2,3],[2,3,4]])
>»   #Matrix subtraction
>»  A-8
Matrix([
[3, 0, -1),
[O, 1,-2)))
```

2) Write Python program to find the sequence function f(x)=x+S,(-S<=x<=S).

```
    for x in range(-5,6):
            y=x+S
            print(y,end='')

012345678910»>
```

3) Using sympy module of python find the eigenvalues and corresponding eigenvectors of the matrix

A=[[4 2 2)(2 4 2)(2 2 4))

Eigenvalues:

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

Q3.a).Attempt any one of the following.

1) Write a Python program to estimate the value of the integral Oto1 1/{1 +x"2)dx using Simpson's (1/3)rd rule(n=4).

```
>»  def simpson13(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+211:f(k)
            else:
                result=result+4""f(k)
        result *= h%3
        return result
```

```
>>> def f{x):
          return 1/(1 +x*x)

>>> simpson13{f,0,1,4)
2.3561764705882355
```

b]1).Write python program to obtained the approximate real root of x"3-4x-9=0 byusing Regula falsi method.

```
def falsePosition(f,x0,x1,e):
          x0=float{x0)
          x1=float{x1)
          e=float(e)
          if f{x0) * f{x1)>0.0:
                print{'Given guess values do not bracket the root.')
                print{'Try Againwith different guess values.')
          else:
                step=1
                condition=True
                while condition:
                      x2=x0-{x1-x0)*f{x0)/(f(x1)-f(x0))
                      print(('lteration %d,x2=%0.6f  and f(x2)=%0.6f%(step,x2,f(x2))
                         if f(x0) * f(x2) < 0:
>>> if f(x0) * f(x2) < 0:
            x1=x2
... else:
            x0=x2
...         step=step+1
... condition=abs(f(x2)) >e
... print ('\nRequired root is:%0.8f%x2)
...
>>>def f(x):
...   return x**3-2*2-5

>>>
```

b]2). Write a python to evaluate interpolate value f(2,2)of the given data f(2)=0.593,f(2.5)=0.816,f(3)=1.078.

```
 interpolate_value = 0.593 + (0.816 - 0.593) * (2 - 2) / (2.5 - 2)
>>> print("The interpolate value of f(2.2) is:", interpolate_value)
The interpolate value of f(2.2) is: 0.593
```

_J

Slip No:15

Q.1-Attempt any two of the following

Q.1.1-Using for loop on python ,find range from 1 to 11 integers.

```
>»   #Generate number between 1 to 11
»>   for i in range(1,11):
           print(i)

1
2
3
4
5
6
7
8
9
10
```

Q.1.2-Use Python code to find,

a)sin75

```
»>   from math import sin
>»   sin(75)
-0.38778163540943045
```

b) Pi/2

```
»>   from numpy import*
»>   print(pi/2)
1.5707963267948966
```

c) e

```
  e
2.718281828459045
```

d)

```
>»   cos(56)
0.8532201077225842
```

Q.1.3-Write Python program to find diameter.area.circumference of the circle with radius is 5.

```
»>   PI=3.14
»>   radius=float(input('Please Enter the radius of a circle:'))
Please Enter the radius of a circle:5
>»   diameter=2*radius
»>   circumference=2*PI*radius
>»   area=PI*radius*radius

»>   print("\nDiameter of a Circal=%.2f"%diameter)

Diameter of a Circal=10.00

>»   print("Circumference of a Circal =%.2f"%circumference)
Circumference of a Circal =31.40

»>   print("Area of a Circal=%.2f"%area)
```

Q.2.1-Using python code construct any three matrices A ,Band C to show that (A+B)+C=A+(B+C)

7

```
>»   from sympy import*
»>  A=Matrix([[1,2,3].[4,5,6],[7,8,9]])
>»  B=Matrix([[1,2,3],[3,4,5],[5,6,7]])
»>  C=Matrix([[2,3,4],[5,6,7],[8,9,111)
>»  (A+B)+C==A+(B+C)
True
>»  #Matrix addition
>>> A+B
Matrix([
[ 2,   4,   6],
[7,    9,11],
[12, 14, 1611)
```

Q.2.2-Using Python ting the eigenvalues and corresponding eigenvectors of the matrix([[3,-2],[6,-4]1)

```
»>  from sympy import*
A»>  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],
[   111)])]
```

Q.2.3 Generate all prime numbers between 1000 to 2000 using python program

```
>»  for num in range(1000,2000):
        if num>1:
            for i in range{2,num):
                if (num % i)==0:
                    break
            else:
                print(num,"is a prime number!")

1009 is a prime number!
1013 is a prime number!
1019 is a prime number!
1021 is a prime number!
1031 is a prime number!
1033 is a prime number!
1039 is a prime number!
1049 is a prime number!
1051 is a prime number!
1061 is a prime number!
1063 is a prime number!
1069 is a prime number!
1087 is a prime number!
1091 is a prime number!
1093 is a prime number!
1097 is a prime number!
1103 is a prime number!
1109 is a prime number!
1117 is a prime number!
1123 is a prime number!
1129 is a prime number!
```

```
1997 is a prime number!
1999 is a prime number!
```

Q.3.A-2-Write python program to estimate a root of an equationf(x)=3x-cos(x)-1 uing Newton Raphson method correct up to four decimal places.

```
>>> import math
>>> def f(x):
        return  3*x-math.cos(x)-1

>>> def derivative_f(x):
        return 3+math.sin(x)
...
>>> def newton_raphson(x):
         h=f(x)/derivative_f(x)
        while abs(h)>=0.0001:
        #x{i+1)=x(i)-f(x)/f'(x)
        x=x-h
        x0=0
        newton_raphson{x0)
>>> print("The value of the root is:",round(x,4))
The value of the root is: 0
```

Q.3.8-1-write python program to obtained the approximate real root of x"3-4x-9=0 by using Regula -falsi method.

```
>> import numpy as np
>>> def func(x):
        return(x**3)-4*x-9

>>> def regula_falsi(a,b):
        if (func(a)*func(b)>=0):
              print("You have not assumed right a and b\n")
              return
        c=a
        while((b-a)>=0.01):
              c=(a*func(b)-b*func(a))/(func(b)-func(a))
              if(func(c)==0.0):
                   break
              elif(func(c)*func(a)<0):
                     b=c
              else:
                     a=c
        print{"The root is:","%.4f"%c)

>>> A=float{input{"Enter the vlue of a:"))
Enter the vlue of a:4
>>> B=float{input{"Enter the value of b:"))
Enter the value of b:2
>>> print{"The root is :","%.4f"% A,B)
The root is : 4.0000 2.0
```

SLIP NO 16

Q.1. Attempt any two of the following.

1. Write Python program to find absolute value of a given real number(n).

Answer

```
int_num = -25

float_num = -10.50
print("The absolute value of an integer number is:", abs(int_num))

The absolute value of an integer number is: 25
print("The absolute value of a float number is:", abs(float_num))

The absolute value of a float number is: 10.5
```

2. Using Python program

List1 = [5, 10, 15, 20, 25, 30] and List2 = [7, 14, 21, 28, 35, 42]

Evaluate

(a) List1 + List2

(b) 7*List1

(c) 11*List2

Answer

```
list1=[5,10,15,20,25,30]
list2=[7,14,21,28,35,42]
list1 + list2
[5, 10, 15, 20, 25, 30, 7, 14, 21, 28, 35, 42]
7*1ist1
[5, 10, 15, 20, 25, 30, 5, 10, 15, 20, 25, 30, 5, 10,
15, 20, 25, 30, 5, 10, 15, 20, 25, 30, 5, 10, 15, 20
```

[5, 10, 15, 20, 25, 30, 5, 10, 15, 20, 25, 30, 5, 10, 15, 20, 25, 30, 5, 10, 15, 20, 25, 30, 5, 10, 15, 20, 25,30,5, 10, 15,20,25,30,5, 10, 15,20,25,30]

11*list2

[7, 14, 21, 28, 35, 42, 7, 14, 21, 28, 35, 42, 7, 14, 21, 28, 35,42, 7, 14, 21, 28, 35,42, 7, 14, 21, 28, 35,42, 7, 14,21,28,35,42, 7, 14,21,28,35,42, 7, 14, 21, 28, 35, 42, 7, 14, 21, 28, 35, 42, 7, 14, 21, 28, 35, 42, 7, 14, 21, 28, 35, 42]

Q.2. Attempt any two of the following. [10]

1. Using Python code, find percentage of marks 70,80, 55, 78, 65 in five subject out of

100 each.

Answer

```
marks = [70, 80, 55, 78, 65]
>>>
>>> percentage = (sum(marks)/500)*100
>>>
>>> print("Percentage of marks:", percentage)
Percentage of marks: 69.6
```

Write python code to find the determinant and inverse of matrices

[1 0 5]

[2 1 6]

[3 40]

andB =

"

[2 5]

```
[2 5]
[-1 4]
```

Answer

```
#import numpy as np
>>>
>>> A = np.array{[[1,0,5],[2,1,6],[3,4,0]])
>>>
>>> detA = np.linalg.det(A)
>>>
>>> print('The determinant of A is", detA)
The determinant of A is 0.9999999999999967
>>>
>>> #Calculate the inverse of A
>>>
>>> invA = np.linalg.inv(A)
>>>
>>> print{"The inverse of A is\n", invA)
The inverse of A is
[[-24.   20.   -5.]
[ 18. -15.     4.]
[   5.   -4.    1.1]
>>>
>>>
>>> #Calculate the determinant of B
>>>
>>> B = np.array{[[2,5],[-1,4]1)
>>>
>>> detB = np.linalg.det{B)
>>>
".>".>'> nrint("Tho rlotorrriin::int r.f R i "  rlotR)
```

```
>>> print("The determinant of B is", detB)
The determinant of B is 13.0
>>>
>>> #Calculate the inverse of B
>>>
>>> invB = np.linalg.inv(B)
>>>
>>> print("The inverse of B is\n", invB)
The inverse of B is
[[ 0.30769231  -0.38461538]
[ 0.07692308     0.15384615]]
>>>
```

Q.3. a. Attempt any one of the following.

1.Write Python program to estimate the value of the integral sin(x)dx using Simpson's(1 /3)rd rule (n=6).

answer

```
>>> # Calculate the value of the integral using Simpson's 1/3 Rule
>>> def simpsons_rule(f, a, b, n):
            h = (b - a)/  n
            s = f(a) + f(b)
            for i in range(1, n, 2):
                  s += 4 * f(a + i * h)
             for i in range (2, n-1, 2):
                  s += 2 * f(a + i * h)
            return (h /  3) * s


#import  math
```

```
#import  math
>>> # Define the function
>>> def f(x):
        return math.sin(x)


>>> # Specify the lower and upper limit of integration
>>>a= 0
>>> b = math.pi
>>>
>>>#Specify the number of intervals
>>> n = 6
>>>
>>> # Calculate and print the value of the integral
>>> print("The value of the integral is", simpsons_rule(f, a, b, n))
The value of the integral is 2.0008631896735363
>>>
```

2. Write Python program to estimate a root of an equation f(x) = x5 + Sx + 1 using Newton-Raphson method in the interval [-1,0].

Answer

```
#Define function
>>> def f(x):
        return 5 + S*x + 1


>>> #Define derivative of function
>>> def f_prime(x):
        return 5
```

```
>>> #Set initial guess
>>> x = -1
>>>
>>> #Set tolerance
>>> tol = 1e-15
>>>
>>> #Implement Newton-Raphson Method
>>> while True:
        x_new = x - f(x)/f_prime(x)
        if abs(x_new - x) < tol:
                break
        x = x_new


>>> #Print result
>>> print("Estimated root of f(x) = xA2 + Sx +1 in
interval [-1,0) is:", x)
Estimated root of f(x) = xA2 + 5x +1 in interval [-
1,0) is: -1.2
```

b. Attempt any one of the following.

2. Write Python program to estimate the value of the integral R 101/(1 +x)dx using Trape-zoidal rule (n=8)

Answer

```
import numpy as np
>>>
>>> def trapezoidal_rule(f, a, b, n):
        h = float(b - a) / n
        s = 0.0
        s += ff a)/ ?.0
```

1,0] is: -1.2

b. Attempt any one of the following.

2. Write Python program to estimate the value of the integral R 101/(1 +**x)dx** using Trape-zoidal rule (n=8)

Answer

```
import numpy as np
>>>
>>> def trapezoidal_rule(f, a, b, n):
          h = float(b - a)/  n
          s = 0.0
          s += f(a) / 2.0
          for i in range(1, n):
                s += f(a  + i * h)
          s += f(b) / 2.0
          returns* h


>>> def f(x):
          return (1 + x)


>>>a= 1
>>> b = 10
>>> n = 8
>>> print("Integral value is", trapezoidal_rule(f, a,
b, n))
Integral value is 58.5
>>>
```

slip no.17

Que.1).attempt any two of the following.

1). Write the python code to print 'Python is bed, and 'python is wonderful', where wonderful is global variable and bad is local variable.

ans:

```
def f():
      global   s
      print(s)
      s = "Only in spring, but london is great as well!"
      print(s)

      s = "I am looking for a coutse in Paris!"
f()
print(s)

I am looking for a coutse in Paris!
Only in spring, but london is great as well!
Only in spring, but london is great as well!
```

3).write python code, find a,b and c such that aA2+bA2=cA2.(where 1<=a,b,c<=50)

```
  ans:- >>> for a in range(1,50):
           for b in range(1,50):
                 c = (a*2 +b*2) *  0.5
                 if a tt  2 + b tt  2 == c tt  2 and c <= 50:
                       print("a:",a,"b:",b,"c:",c)
```

Que.2) Attempt any two of the following.

1).using python code construct any two matrices A and B to show that (AB)A-1=BA-1AA-1.

```
      ANS:->» import numpy as np
>>>
>>> A = np.array([[1,2],[3,4]})
>>> B = np.array([[2,3],[4,5]1)
>>>
>>> A_inverse = np.linalg.inv(A)
>>> B_inverse = np.linalg.inv(B)
>>>
>>> AB_inverse = np.matmul(B_inverse, A_inverse)
>>>
>>> print("A = \n",A)
A=
  [[1 2]
  [3 41]
>>> print("B = \n",B)
B=
  [[2 3]
  [4 51]
>>> print("(A,B)A-1 = \n",AB_inverse)
```

2)using  linsolve codein python solve the follwing system of linear equation

ans:- x-2y+3z=7

2x+y+z=4

```
-3x+2y-2z=-1O
>»from sympy import
»>x,y,z= symbols("x,y,z")
»>  A=matrix([[1,2,3],[2,1,1],[-3,2,-2)))
>»b=matrics([7,-4,-1OJ)
>»linsolve((A,b),[x,y,z])
finteset((z-1,2-2*z,z))
```

Que.3)

a).write any one of the following

1).write a python program to find f{3) to the functional value f(1)=2, f(2)=10, f{4)=68 by using lagrange method.

ans:-#importing libraries

```
»>  import numpy as np
>»  import matplotlib.pyplot as pit
>>>
»>  #defining the function for lagrange
        sum= O
         for i in range(len(x_values)):
                product = y_values[i]
                for j in range{len(x_values)):
                        if j != i:
                product = product*{x - x_valuesU])/(x_values[i] - x_valuesU])
                sum = sum + product
         return sum

»>  #defining the function values
»>  x_values = np.array{[1,2,4])
>»  y_values = np.array{[2,10,68))
>>>
»>  #finding the value of f{3)
»>  f3 = lagrange(3, x_values, y_values)
>>>
```

b).attempt any one of the following.

1).write python program to obtained the approximate real root of x"2-2x-1=O by using Regula -falsi method in the interval [1,3).

```
ans:- »>  def f(x):
         return x*x - 2*x -1

 »>  def regulaFalsi{a,b):
          if f(a) * f{b) >= 0:
                print{"You have not assumed right a and b\n")
                return -1

         c = a # Initialize result

         for i in range{SOO):

                # Find the point that touches x axis
                c = (a*f{b) - b*f{a))/ (f{b) - f(a))

                # Check if the above found point is root
                if f(c) == 0:
                        break
```

_J

```
                    # Decide the side to  repeat the steps
                    elif  f(c)*f(a) < 0:
                            b=c
                    else:
                            a=c
            print("The value of root is : ", '%.4f' %c)

... # Driver Code
... # Initial values assumed
... a = 1
    File "<stdin>", line 26
        a=1
        ^
SyntaxError: invalid syntax
>>> b = 3
>>> reg

>>> print("The value of f(3) is: ",f3)
The value of f(3) is:    32.0
```

_J

slip no18

01

1. use python code to find minimum value from the given numbers

16,3,5,48,2,4,5,6,78,12,5,6,24.

```
ype "help", "copyright", "credits" or "license" for more information.
»> list1=[16,3,5,48,2,4,5,6,78,12,5,6,24)
»>  print ("smallest element is:",list1[01)
smallest element is: 16
```

2. use python code to find hypotenuse of triangle whose sides are 12 and 5

```
from math import sqrt
»>  print("input lenghts of shorter triangle sides:")
input lenghts of shorter triangle sides:
»>  a = float (input("a:"))
a:
Traceback (most recent call last):
   File "<stdin>", line 1, in <module>
ValueError: could not convert string to float:"
»>  a = float (input("a:"))
a:12
>»  b = float(input("b:"))
b:5
»>  c = sqrt(a**2 +b**2)
»>  print ("Te")
Te
»>  he length of the hypotenuse is:",c)
   File "<stdin>", line 1
      he length of the hypotenuse is:",c)
             /\/\/\/\/\/\
SyntaxError: invalid syntax
»>  print("the lenght of the hypotenuse is:",c)
the lenght of the hypotenuse is: 13.0
```

3. use python code to remove all digits  after decimal of the given number 125312.3142

```
»>  number1 = 125312.3142
»>  new_number1 = int(125312.3142)
»>  print ("number1=",125312.3142)
number1= 125372.3142
»>  print(type(l 25312.3142))
<class 'float'>
>»  print(type(l 25312.3142))
<class 'float'>
»>  print(type(125312))
<class 'int'>
>>>
```

02

2.use while code on python to find sum of first twenty natural number

```
»>total= 0
»> fori inrange(1,21):
        total+= i
```

```
... print(total)
210
```

Q3

A

2. write python program to evaluate interpolate value f(2.9) of the given data

```
>>>#data= {2.0: 2.2, 2.5: 2.6, 3.0: 2.9, 3.5: 3.2}
>>>
>>> import scipy.interpolate as interp
>>>
>>>data= {2.0: 2.2, 2.5: 2.6, 3.0: 2.9, 3.5: 3.2}
>>>
>>> x = list(data.keys())
>>> y = list(data.values())
>>>
>>> f = interp.interp1d(x, y, kind='linear')
>>>
>>> x0 = 2.9
>>> y0 = f(x0)
>>>
>>> print("f(2.9) =", y0)
f(2.9 ) = 2.84
```

B

1. write python program to obtained the approximate real root of x"3-5x-9=0 in [2,3] using regula falsi method.

```
>>> # Python program to find the approximate real root of x 3 - 5x - 9 = 0
>>> # in [2,3] using Regula-falsi method
>>>
>>> # Function to find the root
>>> def regulaFalsi(xl, xu):

        # Initialize the root
        root= 0

        # Set the difference between two
        # roots as maximum
        epsilon = 0.01

        # Iterate until root is
        # found within epsilon
        while (abs(xu - xi) > epsilon):

            # Calculate the false position
            root = xu - ((3 - 5*xu - 9)(xl - xu))/(2(xl - 5*xl - 9))

            # Check if xr is root
            if (3 - 5*root - 9 == 0.0):
                break

            # Decide the side to repeat the steps
            if (3 - 5*root - 9)*(3 - 5*xu - 9
```

QUE NO 1

1-WRITE A PYTHON CODE TO DISPLAY MULTIPLICATION TABLES OF NUMBERS 2 TO 10

ANS

```
for i in range(2,11):
        print("Multiplication  Table   of",i)
        for j in range(1,11):
                print(i,"X",j,"=",i*j}
        print("\n"}Multiplication Table  of   2
```

Slip 19

```
OUTPUT
2 X 1 = 2
2 X 2 = 4
2 X 3 = 6
2 X 4 = 8
2 X 5 = 1 0
2 X 6 = 1 2
2 X 7  = 14
2 X 8 = 1 6
2 X 9 = 1 8
2 X 10  = 20


Multiplication Table of 3
3 X 1  = 3
3 X 2 = 6
3 X 3 = 9
3 X 4 = 1 2
3 X 5 = 1 5
3 X 6 = 1 8
3 X 7  = 21
3 X 8 = 24
3 X 9 =  27
3 X 10  = 30


Multiplication Table of 4
4 X 1 = 4
4 X 2 = 8
4 X 3 = 1 2
4 X 4 = 1 6
4 X 5 = 20
4 X 6 = 24
4 X 7  = 28
4 X 8 = 32
4 X 9 = 36
4 X 10  = 40


Multiplication Table of 5
5 X 1 = 5
5 X 2 = 1 0
5 X 3 = 1 5
5 X 4 = 20
```

_J

5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50

Multiplication Table of 6
6 X 1 = 6
6 X 2 = 1 2
6 X 3 = 1 8
6 X 4 = 24
6 X 5 = 30
6 X 6 = 36
6 X 7 = 42
6 X 8 = 48
6 X 9 = 54
6 X 10 = 60

Multiplication Table of 7
7 X 1 = 7
7 X 2 = 14
7 X 3 = 21
7 X 4 = 28
7 X 5 = 35
7 X 6 = 42
7 X 7 = 49
7 X 8 = 56
7 X 9 = 63
7 X 10 = 70

Multiplication Table of 8
8 X 1 = 8
8 X 2 = 1 6
8 X 3 = 24
8 X 4 = 32
8 X 5 = 40
8 X 6 = 48
8 X 7 = 56
8 X 8 = 64
8 X 9 = 72
8 X 10 = 80

Multiplication Table of 9
9 X 1 = 9
9 X 2 = 1 8
9 X 3 = 27
9 X 4 = 36
9 X 5 = 45
9 X 6 = 54
9 X 7 = 63
9 X 8 = 72

7

```
Multiplication Table of 10
10 X 1 = 10
10 X 2 = 20
10 X 3 = 30
10 X 4 = 40
10 X 5 = 50
10 X 6 = 60
10 X 7 = 70
10 X 8 = 80
10 X 9 = 90
10 X 10 = 100
```

3-WRITE PYTHON CODE TO LIST **NAME AND** BIRTH DATE OF 5 STUDENTS IN YOUR CLASS.

```
student_dict = {
      'John':'May 2, 2000',
      'Adam':'June 15, 1999',
      'Sara':'August 23, 1998',
      'Olivia':'November 18, 2001',
      'Emily':'January 12, 2003'
}

for name, birth_date in studenLdict.itemsQ:
      print(f'{name} was born on {birth_date}.')
```

OUTPUT-

```
      John was born on May 2, 2000.
Adam was born on June 15, 1999.
Sara was born on August 23, 1998.
Olivia was born on November 18, 2001.
Emily was born on January 12, 2003.
import math
```

QUE 2

1:-Write python code to find transpose and inverse of matrix

**A=**

122

2 1 2

221

QUE3

A 1-Write Python program to estimate the value of the integral $f$ 1

01

1+x2 dx by using

Simpson's ( 3

8 )th rule (n=6).

**ANS-**

```
def simpsons(f, a, b, n):
      h = (b - a)/ n
      s = f(a) + f(b)

      for i in range(1, n):
```

_J

```
        if i % 2 == O:
            s += 2 * f(a + i * h)
        else:
            s += 4 * f(a + i * h)

    return s * (h / 3)

def f(x):
    return (1 + x * x)

a=O
b=1
n=6

print
```

OUTPUT-

```
<function print>
```

```
def newton_backward(x, x_values, y_values):
    '''
```

QUE-3/B-

2:- Write python program to find sin(42)0 using Newton backward interpolation for mula for the data:

ANS:-

sin300 = 0.5, sin350 = 0.5736, sin400 = 0.6428, sin450 = 0.7071.

```
    Finds the value of the given function f(x) at point x using Newton's backward
    interpolation formula for given set of values.
    Parameters:
    x: The point at which f(x) is to be evaluated.
    x_values: List of values of x.
    y_values: List of values of f(x) corresponding to x_values.
    Returns:
    Value of the function f(x) at point x.
    '''
    n = len(x_values)
```

# Slip 20

_J

Q.1.
1) . Write Python code to print first n natural numbers and their square roots of input integer n.

answer:-

```
n = int(input('Please enter a positive integer: '))

# print first n natural numbers
for i in range(1, n+1):
        print(i, end=' ')
print()

# print square roots of first n natural numbers
for i in range(1, n+1):
        print(i"-""0.5, end='')
print()
```

output-

```
Please enter a positive integer: 6
```

Q.1.
2) Use Python code to find sum of square of first twenty five natural numbers.

ANSWER:-

```
sum_of_squares = O

for i in range(1,26):
    sum_of_squares += i'-""2

print(sum_of_squares)
```

OUTPUT:- 5525

Q.1.
3) Write Python code to find all positive divisors of given number n

ANSWER:-

```
n = int(input("Enter a number: "))

divisors=☐

for i in range(1,n+1):
        if n % i == 0:
                divisors.append(i)

print("The divisors of",n,"are",divisors)
```

OUTPUT:-

7

_J

ENTER A NUMBER-25

Q.2.

1)Write python code to display tuple 'I am Indian ' and the second letter in this tuple

ANSWER:-

```
tuple = ('I', 'am', 'Indian')

print(tuple[1])
```

OUTPUT:-AM

2)Write python code to display the matrix whose all entries are 10 and order is (4,6).

ANSWER:-

```
import numpy as np

matrix = np.full((4,6), 10)
print(matrix)
```

output:

```
[[101010101010]
  [101010101010]
  [101010101010]
  [10 10 10 10 10 1OJ]
```

Q.2.

3) Write Python program to diagonalize the matrix

[
3-2
6-4

ANSWER:-

```
import numpy as np

# Initializing matrix
matrix = np.array([[3, -2], [6, -411)

# Calculate eigenvalues and eigenvectors.
eigenvalues, eigenvectors = np.linalg.eig(matrix)

# Diagonalize the matrix.
diagonalize_matrix = np.diag(eigenvalues)

print("Matrix: \n", matrix, "\n")
print("Eigenvalues: \n", eigenvalues, "\n")
print("Eigenvectors: \n", eigenvectors, "\n")
```

_J

```
print("Diagonalize matrix: \n", diagonalize_matrix, "\n")
```

OUTPUT:-

```
[[ 1.77635684e-15    0.00000000e+00)
  [ 0.00000000e+00 -1.00000000e+00))
```

Q.3

1) Write Python program to estimate the value of the integral J3
1 cos(x)dx using Simp-
son's ( 3
8 )th rule (n=6).

ANSWER:-

```
import math
>>>
>>> def simpsons(f, a, b, n):
            h = (b-a) / n
           s = f(a) + f(b)
           for i in range(1, n, 2):
                  s += 4 * f(a + i * h)
           for i in range(2, n-1, 2):
                  s += 2 * f(a + i * h)
           return (s * h} / 3

>>> def f(x):
           return math.cos(x)

>>> a, b = 3, 1
>>> n = 6
>>>
>>> I = simpsons(f, a, b, n)
>>>
>>> print("The value of the integral is", I)
```

output-

```
The value of the integral is 0.7003996546766208
```

# Slip 21

_J

Q:-1:1

write python code to display multiplication tables of numbers 20 to 30

Answers:-

```
for x in range(20,31):
        print("Table of",x)
        for y in range(1,11):
                print(x,"x",y,"=",x""y)
        print("\n")
```

output:-

range 'x'20,31

range 'y'1,11

Q:1:2

write python code to list name and birth date of 5 students in your class

Answers:-

```
students = I
        {'name': 'Bob', 'birth_date': '01/01/2000'},
        {'name': 'John', 'birth_date': '02/02/2001'},
        {'name': 'Sally', 'birth_date': '03/03/2002'},
        {'name': 'Adam', 'birth_date': '04/04/2003'},
        {'name': 'Jane', 'birth_date': '05/05/2004'}



for student in students:
        print(f"{student['name']}:   {student['birth_date']}")
```

Q:2:1

using python constract the following matrices

1. matrix of order 5*6 with all entries **1**
2. zero matrix of order 27*33
3. identity matrix of order 5

Answers:-

```
1. [[1, 1, 1, 1, 1, 1),
        [1, 1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1, 11]

2. [[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, 0, 0, 0),
        �QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQm
        �QQQQQQQQQQQQQQQQQQQQ
```

Q:2:2

write python code to perform thr R2 + 2R1 row operation on given matrix

R=[[[111]222]333]

_J

Aswers:-

R2 = R2 + 2*R1

R[2] = [R[2][0] + (2*R[1][0]), R[2][1] + (2*R[1][1]), R[2][2] + (2*R[1][2])]

```
R = [1111,222,333],
     [111,222,333],
     [444,666,999]]
```

output:

```
[1 2 3]
[1 2 3]
[1 2 3]
```

Q:3:1

write python program to find the approximate root of the equation x"5+ 3x+1=O, by using newtons raphsons method

Answers:-

```
# Python program to find the approximate root
# of the equation x"5+3x+1=Oby using
# Newton Raphson method

# Initialize the value of xO
xO = 0.5

# Maximum number of iteration
max_itr = 20

# Set relative error to a large number
rel_error = 1e5

# Function f(x)
def f(x):
    return pow(x, 5) + 3*x + 1

# Derivative of f(x)
def df(x):
    return 5*pow(x, 4) + 3

# Iterate till themaximum number of iteration
for i in range(max_itr):

    # Calculate x1 from xO
    x1 = xO - (f(xO)/df(xO))

    # Calculate relative error
    rel_error = abs((x1-x0)/x1)*100

    # Print the current values of xO and x1
    print("Iteration
 output:
 #update xO
```

_J

```
    x0=x1
    # print the root
    print("the approximate root is:",x1)
```

Q:3:B:1

write python program to obtained th eapproximate real root of xsin(x)+cos(x)=0 by using regula -falsi method

Answers:-

```
# Python Program to find the approximate root of
# xsin(x)+cos(x)=0 using Regula - Falsi method

# Function to calculate f{x)
def f{x):
    return x * sin(x) + cos(x)

# Prints root of the equation f{x) = 0
 # in the interval [a, b]
 def regulaFalsi(a, b):
     if f(a) * f{b) >= 0:
          print("You have not assumed right a and b")
          return -1

    c = a # Initialize result

    for i in range{1000):


          # Find the point that touches x axis
          c = {a* f{b) - b * f{a))/ {f{b) - f{a))

          # Check if the above found point is root
          if f{c) == 0:
               break

          # Decide the side to repeat the steps
          elif f(c)
```

```
 output:-
 # main program
 x0=-20
 x1=20
 e=0.01
 regula-falsi{x0,x1,e)
```