Practical 1

```
c1=34+7j
c2=32+27j
print("Addition of two complex number is ", c1+c2)
print("Subtraction of two complex number is ", c1-c2)
print("Multiplication of two complex number is ", c1*c2)
print("Division of two complex number is ", c1/c2)

'''
t=3+4j
print(t)
m=t.conjugate()
print("conjugate of t is  ",m)

'''
import matplotlib.pyplot as plt
x=3+2j
a=[-2+4j,-1+2j,0+2j,1+2j,2+2j,-1+4j,0+4j,1+4j]
A=[x.real for x in a]
B=[x.imag for x in a]
plt.scatter(A,B,color="blue")
plt.show()

'''
import matplotlib.pyplot as plt
s={3+3j,4+3j,2+1j,5+1j,2+1j}
angle=int(input("Enter the angle rotation"))
if angle==90:
    s1={x*1j for x in s}
    print(s1)
    x=[x.real for x in s1]
    y=[x.imag for x in s1]
    plt.plot(x,y,'ro')
    plt.axis([-6,6,-6,6])
    plt.show()
else:
    print("invalid angle")
'''
```

Practical 2

```
import numpy as np
#enter vector as n-list
x=np.array([5,6,7])
y=np.array([1,2,3])
print(x)
print(y)
print("enter value of a and b")
a=int(input())
```

```
b=int(input())
c=a*x+b*y
d=np.dot(x,y)
print("au+bv vector is " , c)
print("dot product is ",d)

Practical 3
'''
import numpy as np
M=np.array([[1,1,1],[3,4,7],[9,6,3]])
M
#matrix M is
print("matrix M is ",M)
Y=M[:,0:1]
Y
#first column of matrix M is
print("first column of matrix M is ",Y)
x=M[:,0:2]
#first two columns of matrix M is
print("first two columns of matrix M is ",x)
t=M[:,0:3]
#all columns of matrix M is
print("all three columns of matrix M is ",t)

'''
import numpy as np
M=np.array([[1,1,1],[3,4,7],[9,6,3]])
M
#matrix M is
print("matrix M is ",M)
a=6
scalar=a*M
print("scalar-matrix multiplication is  ",scalar)

'''
x=[[12,7],[4,5],[3,8]]
t=[[0,0,0],[0,0,0]]
print("original matrix")
print(x)
print("transpose of matrix")
for i in range(len(x)):
        for j in range(len(x[0])):
                t[j][i]=x[i][j]
                for r in t:
                        print(r)
'''

Practical 4

import numpy as np
```

```
x=np.array([1,4,6])
y=np.array([[2,3],[3,4],[4,5]])
print(np.dot(x,y))


'''
import numpy as np
A=np.array([[3,2,2],[4,1,5],[1,2,3]])
print("matrix A is ",A)
B=np.array([[1,2,3],[1,1,1],[2,2,2]])
print("matrix B is ",B)
print("multiplication of two matrices A & B is ")
M=([[0,0,0],[0,0,0],[0,0,0]])
for i in range(len(A)):
        for j in range(len(B[0])):
                for k in range(len(B)):
                        M[i][j]+=A[i][k]*B[k][j]
                        for r in M:
                                print(r)
'''
```

Practical 5

```
import numpy as np
from numpy.linalg import inv
a=np.array([[1,2],[3,4]])
b=inv(a)
print(b)
```

Practical 6

```
from scipy.linalg import lu
import numpy as np
M=np.array([[1,2,3],[3,-1,0],[2,2,2]])
u=lu(M)
print(M)
print(u)
```

Practical 7

```
N=54
print(N)
a=9
b=6
print("factors of N are a and b",a,b)
x=(a+b)/2
y=(a-b)/2
print("x and y is "x,y)
a1=x*x
b1=y*y
```

```
print("a1 and b1 is " a1,b1)
N=a1-b1
print(N)

'''
import math
print("gcd of x & y is :",end="")
print(math.gcd(12,16))
'''

Practical 8
import numpy as np
def oprojection (of_vec,on_vec):
        x1=np.array(of_vec)
        x2=np.array(of_vec)
        scal=np.dot(x2,x1)/np.dot(x1,x2)
        vec=scal*2
        return round(scal,10),np.around(vec,decimals=10)
print(oprojection([2.0,2.0],[1.0,0.0]))
print(oprojection([2.0,2.0],[6.0,2.0]))

Practical 9
import numpy as np
A=np.mat("-2 1;12 -3")
print("A \n",A)
print("eigen values of A are ",np.linalg.eigvals(A))
eigenvalues,eigenvectors=np.linalg.eig(A)
print("first set of eigen values ",eigenvalues)
print("eigen vectors are ",eigenvectors)
```