Question 1

```
import numpy as np
def inner_product():
  x = np.array([2-4j, 0+1j])
  print("Printing First matrix:")
  print(x)

  y = np.array([[2+4j ,0+4j]])
  print("Printing Second matrix:")
  print(y)
  z = np.vdot(x, y)
  print("Inner Product of first and second complex vectors is:",z)



def length():
  x = np.array([2-4j, 0+1j])
  xtx=np.sqrt(np.abs(np.vdot(x, x)))
  print("Length of a complex vector",xtx)

A = np.array([[-1,1], [1,-1]])
eigenvalues, eigenvectors = np.linalg.eig(A)
print("Original Matrix A:")
print(A)
print("\nEigenvalues:")
print(eigenvalues)
print("\nEigenvectors:")
print(eigenvectors)
inner_product()
length()
```

```
Original Matrix A:
[[-1  1]
 [ 1 -1]]

Eigenvalues:
[ 0. -2.]

Eigenvectors:
[[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]
Printing First matrix:
[2.-4.j 0.+1.j]
Printing Second matrix:
[[2.+4.j 0.+4.j]]
Inner Product of first and second complex vectors is: (-8+16j)
Length of a complex vector 4.58257569495584
```

Question 2

```
def diagonalize(A):
    eigenvalues, eigenvectors = np.linalg.eig(A)
    D = np.diag(eigenvalues)
    P = eigenvectors
    P_inv = np.linalg.inv(P)
    A_reconstructed = P @ D @ P_inv
    return A_reconstructed

def stable(eigenvalues):
  for eigenvalue in eigenvalues:
    if eigenvalue<0 and eigenvalue==0:
      return True
    else:
      return False

A = np.array([[1,2],[3,-1]])
B = diagonalize(A);
print(B)
eigenvalues , eigenvectors  = np.linalg.eig(A)
if stable(eigenvalues):
  print("Stable")
else:
  print("Not Stable")
```

```
[[ 1  2]
 [ 3 -1]]
[[ 1.  2.]
 [ 3. -1.]]
Not Stable
```

Question 3

```
import numpy as np
def proj(y,u):
  y= np.array(y)
  u= np.array(u)
  proj_y=(np.dot(y, u)/np.dot(u, u))*u
  return proj_y
def gram_smidth(vectors):
  V=[]
  vectors=np.array(vectors)
  for i in range(len(vectors)):
    temp=vectors[i]
    for j in V:
      temp=temp-proj(vectors[i],j)
    V.append(temp)
  return V

A = np.array([[4, 1, -2], [7, -6, 2], [-3, 2, 8]])
print(A)
print(f" The gram smidth process which give the orthogonal basis are \n{gram_smidth(A)}")
```

```
[[ 4  1 -2]
 [ 7 -6  2]
 [-3  2  8]]
 The gram smidth process which give the orthogonal basis are
[array([ 4,  1, -2]), array([ 3.57142857, -6.85714286,  3.71428571]), array([1.69579288, 3.73074434, 5.25695793])]
```