```python
import numpy as np
import matplotlib.pyplot as plt
import math
import cmath


#F17/102284/2017 - Machines Assignment

#Data given
V_rated = 6.6 * 10**3
Power_load = 3000*10**3
pf = 0.8

#impedances
R1 =0.5
X1 = 1
Z1 = complex(R1,X1)
R2 = 0.4
X2 = 1.2
Z2 = complex(R2,X2)

#load ratios
load1 = .5
load2 = .5

#Currents
theta = math.acos(pf)
i_l =   abs(cmath.rect(150,theta))

#A
#3000kW load at 0.8 lagging pf shared at 50:50
#for generator 1

Pgen1 = load1 * Power_load
Vph = V_rated/math.sqrt(3)

#Calculating the exitation voltage for generatore 1
#finding cos(theta1)
theta1 = math.acos(Pgen1/(3*Vph * i_l))


#finding total current it
it = Power_load/(3*Vph*pf)
i1 = cmath.rect(i_l,theta1)
e_a1 = Vph + (i1 * Z1)

#Line value of EMF
V_line = math.sqrt(3) * abs(e_a1)

#Load angle
alpha1 = np.angle(e_a1, deg =True)


#Calcualting Parameters for Generator 2
Pgen2 = load2 * Power_load
i2 = it - i1
theta2 = math.acos(Pgen2/(3*Vph*abs(i2)))

e_a2 = Vph + i2*Z2

#Line value of EMF
V_line2 = math.sqrt(3) * abs(e_a2)

#Load angle foe generator 2
alpha2 = np.angle(e_a2 , deg = True)

#Calculating Exciation voltage for  generator 1
#e_a2 = i_l * Z2+V_

```

```
#Plotting phasor for Generator 1
#Length of different properties
l_ea1 = abs(e_a1)
l_il = abs(i1)
l_vph1 = abs(Vph)
l_ilRa1 = abs(i1) * R1
l_iljXs1 = abs(i1) * X1

#Il
x_il1 = l_il * math.cos(theta1)
y_il1 = -(l_il * math.sin(theta1))

#Vph
x_Vph1 = l_vph1
y_Vph1 = 0

#Il * Ra
x_ilra1e1 = x_Vph1 + l_il * R1 * math.cos(theta1)
y_ilra1e1 = -(y_Vph1 + l_il * R1 * math.sin(theta1))

#jXsI1
x_jxsend1 = x_ilra1e1 + l_il * X1 * math.sin(theta1)
y_jxsend1 = y_ilra1e1 + l_il * X1 * math.cos(theta1)

#EA
x_ea1end = l_ea1 * math.cos(alpha1)
y_ea1end = l_ea1 * math.sin(alpha1)


#plotting the phasor for Generator 2
#Lengths of different properties
l_ea2 = abs(e_a2)
l_i2 = abs(i2)
l_vph2 = abs(Vph)
l_ilRa2 =abs(i2) * R2
l_iljXs2 = abs(i2) * X2

#Il
x_il2 = l_il * math.cos(theta2)
y_il2 = -(l_il) * math.sin(theta2)

#Vph
x_Vph2 = l_vph2
y_Vph2 = 0

#Il*Ra
x_ilra1e2 = x_Vph2 + l_il * R2 * math.cos(theta2)
y_ilra1e2 = -(y_Vph2 + l_il * R2 * math.sin(theta2))

#JXsIl
x_jxsend2 = x_ilra1e2 + l_i2 * X2 * math.sin(theta2)
y_jxsend2 = y_ilra1e2 + l_il * X2 * math.cos(theta2)


fig = plt.figure(1)
plt.title("Generator 3000kW at 50:50 Phasor Diagram for cos x =0.8 lagging")
#plot for Vph
plt.plot([0, x_Vph1], [0, y_Vph1])
#plot for Ia
plt.plot([0, x_il1], [0, y_il1])
#plot for IaRa
plt.plot([x_Vph1, x_ilra1e1], [y_Vph1, y_ilra1e1])
#plot for jXaIa
plt.plot([x_ilra1e1, x_jxsend1], [y_ilra1e1, y_jxsend1])
#plot for Ea
plt.plot([0, x_jxsend1], [0, y_jxsend1])

#plot for Vph2
plt.plot([0, x_Vph2], [0, y_Vph2])
```

```
#plot for Ia2
plt.plot([0, x_il2], [0, y_il2])
#plot for IaRa
plt.plot([x_Vph2, x_ilra1e2], [y_Vph2, y_ilra1e2])
#plot for jXaIa
plt.plot([x_ilra1e2, x_jxsend2], [y_ilra1e2, y_jxsend2])
#plot for Ea
plt.plot([0, x_jxsend2], [0, y_jxsend2])

plt.grid()
plt.show()
```