```
import numpy as np

chargeX = [1,1,1,1,-1,-1,-1,-1]
chargeY = [1,1,-1,-1,-1,-1,1,1]
chargeZ = [1,-1,1,-1,-1,1,-1,1]

point1 = [0.01,0,0]
point2 = [0.01,0.01,0]
point3 = [0.01,0.01,0.01]
point4 = [0,0,0]

# method to calculate total E field in square given a source point
def total_E_field(field_coord):
        #field_coord should be an array of form (x,y,z)
        Ex = 0
        Ey = 0
        Ez = 0

        for i in range(0,8):
                charge = [chargeX[i],chargeY[i],chargeZ[i]]

                # calculate 3d & 2d distance and Exyz field
                distance2d = np.linalg.norm(np.array(field_coord[0:2]) -
                    np.array(charge[0:2]))
                distance3d = np.linalg.norm(np.array(field_coord) -
                    np.array(charge))
                Exyz = 1./(distance3d*distance3d)

                # calculate z component of E field
                Ez += (np.abs(field_coord[2]-charge[2])/distance3d)*Exyz

                # calculate horizontal plane field
                Exy = (distance2d/distance3d)*Exyz

                # calculate theta2 to help find Ex,Ey
                theta2 = np.arctan(np.abs(field_coord[0] -
                    charge[0])/np.abs(field_coord[1] - charge[1]))

                # calculate the x and y component of the E field
                Ey += np.cos(theta2)*Exy
                Ex += np.sin(theta2)*Exy

        return Ex,Ey,Ez

# call method and print out results
Ex1,Ey1,Ez1 = total_E_field(point1)
Ex2,Ey2,Ez2 = total_E_field(point2)
```

```
Ex3,Ey3,Ez3 = total_E_field(point3)
Exc,Eyc,Ezc = total_E_field(point4)

print('E1 = ',Ex1,Ey1,Ez1)
print('E2 = ',Ex2,Ey2,Ez2)
print('E3 = ',Ex3,Ey3,Ez3)
print('E4 = ',Exc,Eyc,Ezc)
```

Output after running,

```
('E1 = ', 1.5394980763658117, 1.5396520335862838, 1.5396520335862833)
('E2 = ', 1.539549396388431, 1.5395493963884312, 1.5397033707155778)
('E3 = ', 1.539600737801047, 1.539600737801047, 1.539600737801047)
('E4 = ', 1.5396007178390023, 1.5396007178390025, 1.5396007178390023)
```