```python
In [1]:  from math import sqrt, radians
         import matplotlib.pyplot as plt
         from mpl_toolkits.mplot3d import Axes3D
         import numpy as np

         def plot_this_3D_struct(XM):
             ax.scatter(X[0, :], X[1, :], X[2, :], c='r')
             ax.scatter(XM[0, 0], XM[1, 0], XM[2, 0], c='r', marker="x", s=50)

             ax.plot([XM[0][0], XM[0][1]], [XM[1][0], XM[1][1]])
             ax.plot([XM[0][1], XM[0][2]], [XM[1][1], XM[1][2]])
             ax.plot([XM[0][2], XM[0][3]], [XM[1][2], XM[1][3]])
             ax.plot([XM[0][3], XM[0][0]], [XM[1][3], XM[1][0]])

             ax.plot([XM[0][4], XM[0][5]], [XM[1][4], XM[1][5]])
             ax.plot([XM[0][5], XM[0][6]], [XM[1][5], XM[1][6]])
             ax.plot([XM[0][6], XM[0][7]], [XM[1][6], XM[1][7]])
             ax.plot([XM[0][7], XM[0][4]], [XM[1][7], XM[1][4]])

             ax.plot([XM[0][2], XM[0][6]], [XM[1][2], XM[1][6]])
             ax.plot([XM[0][0], XM[0][4]], [XM[1][0], XM[1][4]])
             ax.plot([XM[0][3], XM[0][7]], [XM[1][3], XM[1][7]])
             ax.plot([XM[0][1], XM[0][5]], [XM[1][1], XM[1][5]])

             ax.plot([XM[0][1], XM[0][8]], [XM[1][1], XM[1][8]])
             ax.plot([XM[0][2], XM[0][8]], [XM[1][2], XM[1][8]])
             ax.plot([XM[0][5], XM[0][8]], [XM[1][5], XM[1][8]])
             ax.plot([XM[0][6], XM[0][8]], [XM[1][6], XM[1][8]])


         p = [[0,0,0],[1,0,0],[1,0,1],[0,0,1],[0,1,0],[1,1,0],[1,1,1],[0,1,1]]

         X = np.array([[0,0,0],[1,0,0],[1,0,1],[0,0,1],[0,1,0],[1,1,0],[1,1,1],[0,1,1],[1.5,

         XM = X.transpose()

         # Camera parameters with translation vector, focal length and theata
         # Converting the theta from degrees to radians
         T = np.array([0, 1, -10])  # Translation vector
         f = 5  # Focal length
         theta_deg = 30  # Rotation angle around z-axis
         theta_rad = radians(theta_deg)

         # Rotational matrix along the z-axis
         Rz = np.array([[np.cos(theta_rad), -np.sin(theta_rad), 0],
                        [np.sin(theta_rad), np.cos(theta_rad), 0],
                        [0, 0, 1]])

         # Transform 3D points to camera coordinates
         # subtract the translation vector from each column vector in XM
         X_camera = Rz @ (XM - T.reshape(-1, 1))

         # Perspective projection
```

```
X_image = f * X_camera[:2, :] / X_camera[2, :]
```

```python
# plotting 3D cube https://likegeeks.com/3d-plotting-in-python/
fig = plt.figure(figsize = (10,10))
ax = plt.axes(projection='3d')
ax.grid()
plot_this_3D_struct(XM)
ax.view_init(elev=90., azim=90)

plt.show()
```