ANSWER TO THE QUSTION NO. 5(b)

In [2]:
```
#import libraries
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
t = np.arange(0, 15, 0.1)
t
```

Out[3]:
```
array([ 0. ,  0.1,  0.2,  0.3,  0.4,  0.5,  0.6,  0.7,  0.8,  0.9,  1. ,
        1.1,  1.2,  1.3,  1.4,  1.5,  1.6,  1.7,  1.8,  1.9,  2. ,  2.1,
        2.2,  2.3,  2.4,  2.5,  2.6,  2.7,  2.8,  2.9,  3. ,  3.1,  3.2,
        3.3,  3.4,  3.5,  3.6,  3.7,  3.8,  3.9,  4. ,  4.1,  4.2,  4.3,
        4.4,  4.5,  4.6,  4.7,  4.8,  4.9,  5. ,  5.1,  5.2,  5.3,  5.4,
        5.5,  5.6,  5.7,  5.8,  5.9,  6. ,  6.1,  6.2,  6.3,  6.4,  6.5,
        6.6,  6.7,  6.8,  6.9,  7. ,  7.1,  7.2,  7.3,  7.4,  7.5,  7.6,
        7.7,  7.8,  7.9,  8. ,  8.1,  8.2,  8.3,  8.4,  8.5,  8.6,  8.7,
        8.8,  8.9,  9. ,  9.1,  9.2,  9.3,  9.4,  9.5,  9.6,  9.7,  9.8,
        9.9, 10. , 10.1, 10.2, 10.3, 10.4, 10.5, 10.6, 10.7, 10.8, 10.9,
       11. , 11.1, 11.2, 11.3, 11.4, 11.5, 11.6, 11.7, 11.8, 11.9, 12. ,
       12.1, 12.2, 12.3, 12.4, 12.5, 12.6, 12.7, 12.8, 12.9, 13. , 13.1,
       13.2, 13.3, 13.4, 13.5, 13.6, 13.7, 13.8, 13.9, 14. , 14.1, 14.2,
       14.3, 14.4, 14.5, 14.6, 14.7, 14.8, 14.9])
```

In [4]:
```
len(t)
```

Out[4]:
```
150
```

In [5]:
```
# Declare Final time T
T= 15
Tsq = np.power(T,2)
Tcb = np.power(T,3)
```

In [6]:
```
#initialized A
A = np.array([[1, 0, 0, 0,  0, 0, 0, 0],
              [0, 1, 0, 0,  0, 0, 0, 0],
              [0, 0, 0, 0,  1, 0, 0, 0],
              [0, 0, 0, 0,  0, 1, 0, 0],
              [1, T, Tsq, Tcb,  0, 0, 0, 0],
              [0, 1, 2*T, 3*Tsq,  0, 0, 0, 0],
              [0, 0, 0, 0,  1, T, Tsq, Tcb,],
              [0, 0, 0, 0,  0, 1, 2*T, 3*Tsq]
              ]
             )
A
```

Out[6]:
```
array([[   1,    0,    0,    0,    0,    0,    0,    0],
       [   0,    1,    0,    0,    0,    0,    0,    0],
       [   0,    0,    0,    0,    1,    0,    0,    0],
       [   0,    0,    0,    0,    0,    1,    0,    0],
       [   1,   15,  225, 3375,    0,    0,    0,    0],
       [   0,    1,   30,  675,    0,    0,    0,    0],
       [   0,    0,    0,    0,    1,   15,  225, 3375],
       [   0,    0,    0,    0,    0,    1,   30,  675]])
```

In [7]:
```
# A pesudo inverse
Ainv = np.linalg.pinv(A)
Ainv
```

Out[7]:
```
array([[ 1.00000000e+00,  6.58070820e-13,  0.00000000e+00,
         0.00000000e+00, -8.64325972e-16,  3.91841507e-15,
         0.00000000e+00,  0.00000000e+00],
       [ 4.16333634e-17,  1.00000000e+00,  0.00000000e+00,
         0.00000000e+00, -1.00180281e-16,  1.21430643e-17,
         0.00000000e+00,  0.00000000e+00],
       [-1.33333333e-02, -1.33333333e-01,  0.00000000e+00,
         0.00000000e+00,  1.33333333e-02, -6.66666667e-02,
         0.00000000e+00,  0.00000000e+00],
       [ 5.92592593e-04,  4.44444444e-03,  0.00000000e+00,
         0.00000000e+00, -5.92592593e-04,  4.44444444e-03,
         0.00000000e+00,  0.00000000e+00],
       [ 4.38264820e-17, -1.56199568e-17,  1.00000000e+00,
         4.13738488e-13, -4.38264820e-17,  2.19459069e-16,
         3.73236598e-16, -9.14524532e-16],
       [-3.59628264e-18, -8.65405541e-17, -1.66533454e-16,
         1.00000000e+00,  3.59628264e-18, -1.80076414e-17,
        -3.85975973e-17,  7.75421394e-16],
       [-1.03997980e-19,  1.18523646e-17, -1.33333333e-02,
        -1.33333333e-01,  1.03997980e-19, -5.20842611e-19,
         1.33333333e-02, -6.66666667e-02],
       [ 9.93178042e-21, -4.00814858e-19,  5.92592593e-04,
         4.44444444e-03, -9.93178042e-21,  4.97355363e-20,
        -5.92592593e-04,  4.44444444e-03]])
```

In [8]:
```
#initialized b
b =  np.array([[0],#x1(0)
               [0],#1
               [0],#x3(0)
               [-0.5],#x2(0)
               [5],#x1(T)
               [0],#1
               [5],#x3(T)
               [-0.5] #x2(T)
              ]
             )

b
```

Out[8]:
```
array([[ 0. ],
       [ 0. ],
       [ 0. ],
       [-0.5],
       [ 5. ],
       [ 0. ],
       [ 5. ],
       [-0.5]])
```

In [9]:
```
#matrix multiplication x = Ainv * b
x= np.matmul(Ainv, b)
x
```

Out[9]:
```
array([[-4.32162986e-15],
       [-5.00901404e-16],
       [ 6.66666667e-02],
       [-2.96296296e-03],
       [-2.04764931e-13],
       [-5.00000000e-01],
       [ 1.66666667e-01],
       [-7.40740741e-03]])
```

In [10]:
```
a11 = x[0]
a11
```

Out[10]:
```
array([-4.32162986e-15])
```

In [11]:
```
a12 = x[1]
a12
```

Out[11]:
```
array([-5.00901404e-16])
```

In [12]:
```
a13 = x[2]
a13
```

Out[12]:
```
array([0.06666667])
```

In [13]:
```
a14 = x[3]
a14
```

Out[13]:
```
array([-0.00296296])
```

In [14]:
```
a21 = x[4]
a21
```

Out[14]:
```
array([-2.04764931e-13])
```

In [15]:
```
a22 = x[5]
a22
```

Out[15]:
```
array([-0.5])
```

In [16]:
```
a23 = x[6]
a23
```

Out[16]:
```
array([0.16666667])
```

In [17]:
```
a24 = x[7]
a24
```

Out[17]:
```
array([-0.00740741])
```

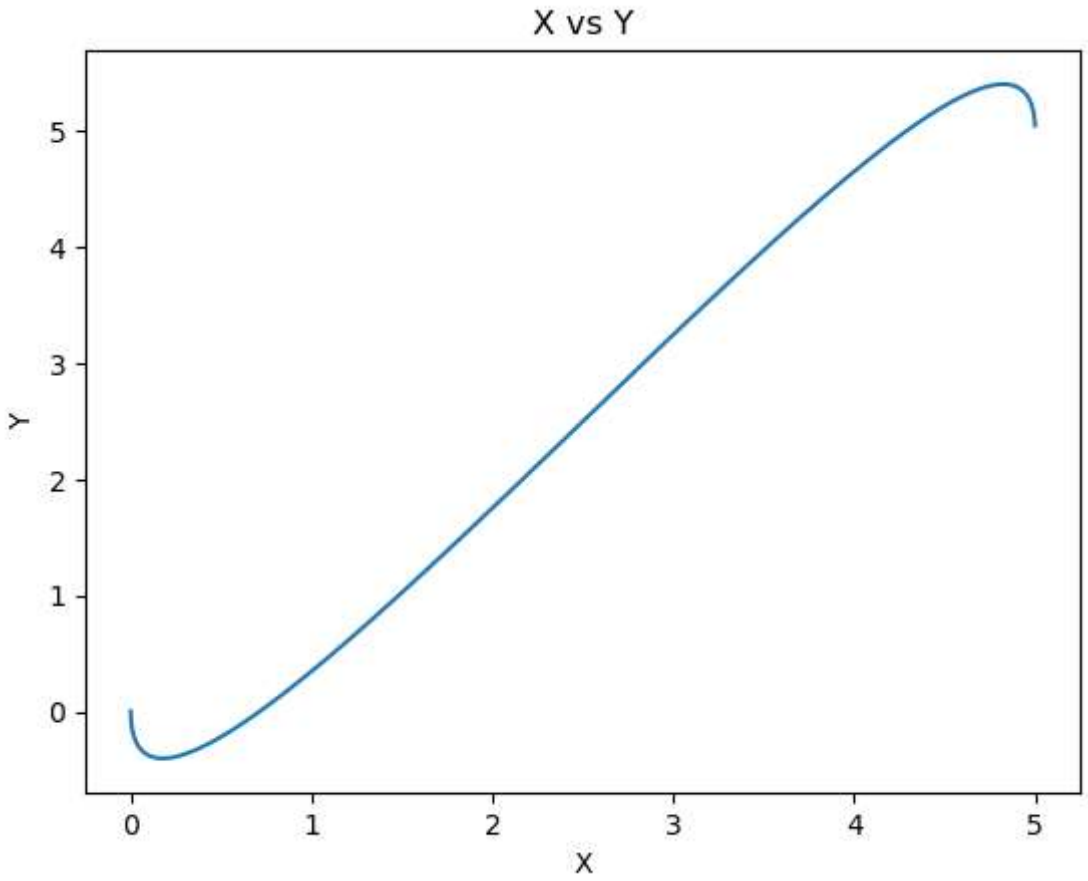
In [18]:
```
for i in t:
  X = a11 + (a12* t) + (a13 * np.power(t,2)) + (a14 * np.power(t,3))
  Y = a21 + (a22* t) + (a23 * np.power(t,2)) + (a24 * np.power(t,3) )

X, Y
```

Out[18]:
```
(array([-4.32162986e-15,  6.63703704e-04,  2.64296296e-03,  5.92000000e-03,
         1.04770370e-02,  1.62962963e-02,  2.33600000e-02,  3.16503704e-02,
         4.11496296e-02,  5.18400000e-02,  6.37037037e-02,  7.67229630e-02,
         9.08800000e-02,  1.06157037e-01,  1.22536296e-01,  1.40000000e-01,
         1.58530370e-01,  1.78109630e-01,  1.98720000e-01,  2.20343704e-01,
         2.42962963e-01,  2.66560000e-01,  2.91117037e-01,  3.16616296e-01,
         3.43040000e-01,  3.70370370e-01,  3.98589630e-01,  4.27680000e-01,
         4.57623704e-01,  4.88402963e-01,  5.20000000e-01,  5.52397037e-01,
         5.85576296e-01,  6.19520000e-01,  6.54210370e-01,  6.89629630e-01,
         7.25760000e-01,  7.62583704e-01,  8.00082963e-01,  8.38240000e-01,
         8.77037037e-01,  9.16456296e-01,  9.56480000e-01,  9.97090370e-01,
         1.03826963e+00,  1.08000000e+00,  1.12226370e+00,  1.16504296e+00,
         1.20832000e+00,  1.25207704e+00,  1.29629630e+00,  1.34096000e+00,
         1.38605037e+00,  1.43154963e+00,  1.47744000e+00,  1.52370370e+00,
         1.57032296e+00,  1.61728000e+00,  1.66455704e+00,  1.71213630e+00,
         1.76000000e+00,  1.80813037e+00,  1.85650963e+00,  1.90512000e+00,
         1.95394370e+00,  2.00296296e+00,  2.05216000e+00,  2.10151704e+00,
         2.15101630e+00,  2.20064000e+00,  2.25037037e+00,  2.30018963e+00,
         2.35008000e+00,  2.40002370e+00,  2.45000296e+00,  2.50000000e+00,
         2.54999704e+00,  2.59997630e+00,  2.64992000e+00,  2.69981037e+00,
         2.74962963e+00,  2.79936000e+00,  2.84898370e+00,  2.89848296e+00,
         2.94784000e+00,  2.99703704e+00,  3.04605630e+00,  3.09488000e+00,
         3.14349037e+00,  3.19186963e+00,  3.24000000e+00,  3.28786370e+00,
         3.33544296e+00,  3.38272000e+00,  3.42967704e+00,  3.47629630e+00,
         3.52256000e+00,  3.56845037e+00,  3.61394963e+00,  3.65904000e+00,
         3.70370370e+00,  3.74792296e+00,  3.79168000e+00,  3.83495704e+00,
         3.87773630e+00,  3.92000000e+00,  3.96173037e+00,  4.00290963e+00,
         4.04352000e+00,  4.08354370e+00,  4.12296296e+00,  4.16176000e+00,
         4.19991704e+00,  4.23741630e+00,  4.27424000e+00,  4.31037037e+00,
         4.34578963e+00,  4.38048000e+00,  4.41442370e+00,  4.44760296e+00,
         4.48000000e+00,  4.51159704e+00,  4.54237630e+00,  4.57232000e+00,
         4.60141037e+00,  4.62962963e+00,  4.65696000e+00,  4.68338370e+00,
         4.70888296e+00,  4.73344000e+00,  4.75703704e+00,  4.77965630e+00,
         4.80128000e+00,  4.82189037e+00,  4.84146963e+00,  4.86000000e+00,
         4.87746370e+00,  4.89384296e+00,  4.90912000e+00,  4.92327704e+00,
         4.93629630e+00,  4.94816000e+00,  4.95885037e+00,  4.96834963e+00,
         4.97664000e+00,  4.98370370e+00,  4.98952296e+00,  4.99408000e+00,
         4.99735704e+00,  4.99933630e+00]),
 array([-2.04764931e-13, -4.83407407e-02, -9.33925926e-02, -1.35200000e-01,
        -1.73807407e-01, -2.09259259e-01, -2.41600000e-01, -2.70874074e-01,
        -2.97125926e-01, -3.20400000e-01, -3.40740741e-01, -3.58192593e-01,
        -3.72800000e-01, -3.84607407e-01, -3.93659259e-01, -4.00000000e-01,
        -4.03674074e-01, -4.04725926e-01, -4.03200000e-01, -3.99140741e-01,
        -3.92592593e-01, -3.83600000e-01, -3.72207407e-01, -3.58459259e-01,
        -3.42400000e-01, -3.24074074e-01, -3.03525926e-01, -2.80800000e-01,
        -2.55940741e-01, -2.28992593e-01, -2.00000000e-01, -1.69007407e-01,
        -1.36059259e-01, -1.01200000e-01, -6.44740741e-02, -2.59259259e-02,
         1.44000000e-02,  5.64592593e-02,  1.00207407e-01,  1.45600000e-01,
         1.92592593e-01,  2.41140741e-01,  2.91200000e-01,  3.42725926e-01,
         3.95674074e-01,  4.50000000e-01,  5.05659259e-01,  5.62607407e-01,
         6.20800000e-01,  6.80192593e-01,  7.40740741e-01,  8.02400000e-01,
         8.65125926e-01,  9.28874074e-01,  9.93600000e-01,  1.05925926e+00,
         1.12580741e+00,  1.19320000e+00,  1.26139259e+00,  1.33034074e+00,
         1.40000000e+00,  1.47032593e+00,  1.54127407e+00,  1.61280000e+00,
         1.68485926e+00,  1.75740741e+00,  1.83040000e+00,  1.90379259e+00,
         1.97754074e+00,  2.05160000e+00,  2.12592593e+00,  2.20047407e+00,
         2.27520000e+00,  2.35005926e+00,  2.42500741e+00,  2.50000000e+00,
         2.57499259e+00,  2.64994074e+00,  2.72480000e+00,  2.79952593e+00,
         2.87407407e+00,  2.94840000e+00,  3.02245926e+00,  3.09620741e+00,
         3.16960000e+00,  3.24259259e+00,  3.31514074e+00,  3.38720000e+00,
         3.45872593e+00,  3.52967407e+00,  3.60000000e+00,  3.66965926e+00,
         3.73860741e+00,  3.80680000e+00,  3.87419259e+00,  3.94074074e+00,
         4.00640000e+00,  4.07112593e+00,  4.13487407e+00,  4.19760000e+00,
         4.25925926e+00,  4.31980741e+00,  4.37920000e+00,  4.43739259e+00,
```

```
4.49434074e+00,  4.55000000e+00,  4.60432593e+00,  4.65727407e+00,
4.70880000e+00,  4.75885926e+00,  4.80740741e+00,  4.85440000e+00,
4.89979259e+00,  4.94354074e+00,  4.98560000e+00,  5.02592593e+00,
5.06447407e+00,  5.10120000e+00,  5.13605926e+00,  5.16900741e+00,
5.20000000e+00,  5.22899259e+00,  5.25594074e+00,  5.28080000e+00,
5.30352593e+00,  5.32407407e+00,  5.34240000e+00,  5.35845926e+00,
5.37220741e+00,  5.38360000e+00,  5.39259259e+00,  5.39914074e+00,
5.40320000e+00,  5.40472593e+00,  5.40367407e+00,  5.40000000e+00,
5.39365926e+00,  5.38460741e+00,  5.37280000e+00,  5.35819259e+00,
5.34074074e+00,  5.32040000e+00,  5.29712593e+00,  5.27087407e+00,
5.24160000e+00,  5.20925926e+00,  5.17380741e+00,  5.13520000e+00,
5.09339259e+00,  5.04834074e+00]))
```

In [19]:
```
plt.plot(X,Y)
plt.title('X vs Y')
plt.xlabel('X')
plt.ylabel('Y')
```

Out[19]:
```
Text(0, 0.5, 'Y')
```



In [ ]: