

```

import numpy as np
import sys

# Reading number of unknowns
n = 3

# Making numpy array of n x n+1 size and initializing
# to zero for storing augmented matrix
a = np.zeros((n,n+1))

# Making numpy array of n size and initializing
# to zero for storing solution vector
x = np.zeros(n)

# Reading augmented matrix coefficients
# a = [[1,1,1,1],[1,2,4,-1],[1,3,9,1]]
a = [[3,4,3,10],[1,5,-1,7],[6,3,7,15]]

# Applying Gauss Elimination
for i in range(n):
    if a[i][i] == 0.0:
        sys.exit('Divide by zero detected!')
    if a[i][i] != 1:
        divide = a[i][i]
        for col in range(n+1):
            a[i][col] = a[i][col] / divide
    for j in range(i+1, n):
        ratio = a[j][i]/a[i][i]
        for k in range(n+1):
            a[j][k] = a[j][k] - ratio * a[i][k]
        #Printing the Matrix
        for row in a:
            print(row)
        print()
# Back Substitution
x[n-1] = a[n-1][n]/a[n-1][n-1]

for i in range(n-2,-1,-1):
    x[i] = a[i][n]

    for j in range(i+1,n):
        x[i] = x[i] - a[i][j]*x[j]

    x[i] = x[i]/a[i][i]

# Displaying solution
print('\nRequired solution is: ')
for i in range(n):
    print('X%d = %.2f' %(i,x[i]), end = '\t')
```

```

[1.0, 1.3333333333333333, 1.0, 3.3333333333333335]
[0.0, 3.6666666666666667, -2.0, 3.6666666666666665]
[6, 3, 7, 15]

[1.0, 1.3333333333333333, 1.0, 3.3333333333333335]
[0.0, 3.6666666666666667, -2.0, 3.6666666666666665]
[0.0, -5.0, 1.0, -5.0]

[1.0, 1.3333333333333333, 1.0, 3.3333333333333335]
[0.0, 1.0, -0.5454545454545454, 0.9999999999999999]
[0.0, 0.0, -1.727272727272727, -8.881784197001252e-16]
```

```

Required solution is:
X0 = 2.00      X1 = 1.00      X2 = 0.00
```

