```
"""
Registration: xxxx;
Description: (a) Kirchoff Law
Input 3 equations: (R1+R2+R3     -R3         -R2    )(I1) = (0 )
                   (-R3        R3+R4+R5      -R5    )(I2) = (E1)
                   (-R2          -R5       R2+R5+R6 )(I3) = (E2)
Author: AKB
"""

import numpy as np

# input initial condition
n = int(input('Enter number of equations to solve: '))
R1, R4, E1 = input('Enter Resistance R1, R4 in Ohm and voltage E1 in Volt : ').split()
print(R1, R4, E1)
R1 = float(R1); R4 = float(R4); R3 = R2 = float(R1); R5 = R6 = float(R4);
E1 = float(E1); E2 = float(E1)

# Assemble the Matrix-vector combination
R = np.array([[R1+R2+R3,-R3,-R2], [-R3,R3+R4+R5,-R5],[-R2,-R5,R2+R5+R6]])
E = np.array([0,E1,E2])

# Logical case switch Gaussian Elimination & Gauss-Seidel
Gauelim = 1; Gauseid = 0;

# Print Solution using direct solver to match
print ('Using Linear Solver : ',  np.linalg.solve(R,E))
print ('Using Inverse Solver : ', np.dot(np.linalg.inv(R),E))

if(Gauelim):

   print ('Using Gaussian Elimination')
   # Elimination Stage
   for k in range (0, n-1):
       for i in range (k+1,n):
           if R[i,k] != 0.0:
              factor     = R[i,k]/R[k,k]
              R[i,k+1:n] = R[i,k+1:n] - factor*R[k,k+1:n]
              E[i]       = E[i] - factor*E[k]

   # Back Substitution
   for k in range(n-1,-1,-1):
       E[k] = (E[k] - np.dot(R[k,k+1:n],E[k+1:n]))/R[k,k]

   # Print solution
   print ('Current values in Ampere are ', E)

elif(Gauseid):

   print ('Using Gauss-Seidel Iteration')
   x     = np.array([1.0, 1.0, 1.0]) # Guess value
   error = 100         # Initialize with a Guess error
   tol   = 1E-4        # Tolerance
   L     = np.tril(R)  # Lower Triangular matrix
   U     = R - L       # Upper Triangular matrix

   # Iteration
   while error > tol:
      temp = np.dot(np.linalg.inv(L), E-np.dot(U,x))
      error = sum(abs(x - temp))
      if error > tol:
         x = temp

   # Print solution
   print ('Current values in Ampere are ', x)
```

```
"""
Results:
Enter number of equations to solve: 3
Enter Resistance R1, R4 in Ohm and voltage E1 in Volt : 1 2 3
Using Linear Solver :   [ 0.85714286  1.28571429  1.28571429]
Using Inverse Solver :  [ 0.85714286  1.28571429  1.28571429]
Using Gaussian Elimination
Current values in Ampere are  [ 0.85714286  1.28571429  1.28571429]
Using Gauss-Seidel Iteration
Current values in Ampere are  [ 0.85710002  1.28568473  1.2856939 ]
"""
```