import numpy as np

#Part1

def augMatrix(A, b):

r, c = np.shape(A)

Au = np.zeros((r, r+1), dtype='f')

Au[0:r, 0:r] = A

Au[:, r] = b

return Au

#part2

def colfun(Au, j):

r,c=np.shape(Au)

Au[j, j:] = (1. / Au[j, j]) \* Au[j, j:]

for i in range(j + 1, r):

Au[i, :] = Au[i, :] - Au[i, j] \* Au[j,:]

return Au

#part3

A=np.array([[2,3,4],[2,2,6],[2,3,7]],dtype='f')

b=np.array([2,1,6])

Au=augMatrix(A,b)

print("Augmented Matrix:")

print(Au)

#part4

r,c=np.shape(A)

for i in range(r):

Au=colfun(Au,i)

#part5

def backsubstitution(Au,r):

x=np.zeros(r)

for i in range(r-1,-1,-1):

sum=0

for j in range(i+1,r):

sum+=Au[i,j]\*x[j]

x[i]=Au[i,r]-sum

return x

#part6

x=backsubstitution(Au,r)

print(x)