```python
import matplotlib.pyplot as plt
from scipy import signal
import numpy as np
#%%
"""
Q -

x[n]=delta[n+2]+delta[n-1]+delta[n-2]
y[n]=-delta[n-1]+2delta[n-2]
"""

Vlen = 10 #len vector
#plot stem with color
def stem_plot(n,val,color):
    markerline1, stemlines1, baseline1 = plt.stem(n,val)
    plt.setp(markerline1, 'markerfacecolor', color)
    plt.setp(stemlines1, linestyle="-", color=color, linewidth=2 )


#create Delta
def dirac(val=0):
    zero = Vlen/2
    return signal.unit_impulse(Vlen,int(zero-val))

def plotStem(title,ylabel,xlabel,color,x,y):
    plt.figure()
    plt.title(title)
    plt.ylabel(ylabel)
    plt.xlabel(xlabel)
    stem_plot(x,y,color)
    plt.show()

ncov =  np.arange(-10,9,1)
n = np.arange(-5,5,1)
delta  = signal.unit_impulse(8)
x = dirac(2)+dirac(-1)+dirac(-2) # x[n]=delta[n+2]+delta[n-1]+delta[n-2]
y = -dirac(-1)+2*dirac(-2) # y[n]=-delta[n-1]+2delta[n-2]
plotStem("x[n]","amplitude","n",'red',n,x)
plotStem("y[n]","amplitude","n",'green',n,y)
convXY = np.convolve(x,y)
plotStem("conv(x*y)","amplitude","n",'blue',ncov,convXY)
#%%
"""
Q - 7

x=[1 -0.5 0.5 1];
h=[-1 -1 0 0];

"""
#function

def padded_zeros(arr,N): # padded zeros
    padded_array = np.zeros(N)
    padded_array[:len(arr)]+=arr
    return padded_array


#cycle Covolation
def ccov(x,y):
    #check which array more large and pandded with zeros the others
    N=len(x)
    if len(x)>len(y):
        N=len(x)
        y = padded_zeros(y,N)
    else:
        N=len(y)
        x = padded_zeros(x,N)
```

```
68        h = np.zeros(N) # check h[n] vector
69        for n in range(N):
70            for k in range(N):
71                h[n]+=y[k]*x[(n-k)%N]
72        return h
73
74    n = np.arange(0,4,1)
75    x=np.array([1,-0.5,0.5,1])
76    h=np.array([-1,-1,0,0])
77    y = ccov(x,h) # cycle convolation
78    plotStem("x[n]","amplitude","n",'red',n,x)
79    plotStem("h[n]","amplitude","n",'red',n,h)
80    plotStem("y[n]","amplitude","n",'red',n,y)
81
82
83
84
85
```