In [1]:
```python
import numpy as np
import matplotlib.pyplot as plt
import math as math
import cmath as cm
from scipy.signal import freqz
```

Question 1 Part a

In [2]:
```python
n = 10   #Time points
nf = 40 #Frequency points

w_start = -math.pi
w_end = math.pi

x = [1, 2, 3, 4, 5, 10, 1, 2, 9, 5]
print(x)
freqPts = np.arange(w_start, w_end, (w_end - w_start)/nf)
freqPts
```

```
[1, 2, 3, 4, 5, 10, 1, 2, 9, 5]
```

Out[2]:
```
array([-3.14159265, -2.98451302, -2.82743339, -2.67035376, -2.51327412,
       -2.35619449, -2.19911486, -2.04203522, -1.88495559, -1.72787596,
       -1.57079633, -1.41371669, -1.25663706, -1.09955743, -0.9424778 ,
       -0.78539816, -0.62831853, -0.4712389 , -0.31415927, -0.15707963,
        0.        ,  0.15707963,  0.31415927,  0.4712389 ,  0.62831853,
        0.78539816,  0.9424778 ,  1.09955743,  1.25663706,  1.41371669,
        1.57079633,  1.72787596,  1.88495559,  2.04203522,  2.19911486,
        2.35619449,  2.51327412,  2.67035376,  2.82743339,  2.98451302])
```

In [3]:
```python
def dtft(x,freqPts):
    output = [0]*freqPts.__len__()
    for k in range(freqPts.__len__()):
        for t in range(n):
            output[k] += x[t] * cm.exp(-1j * freqPts[k] * t)
    return output
```

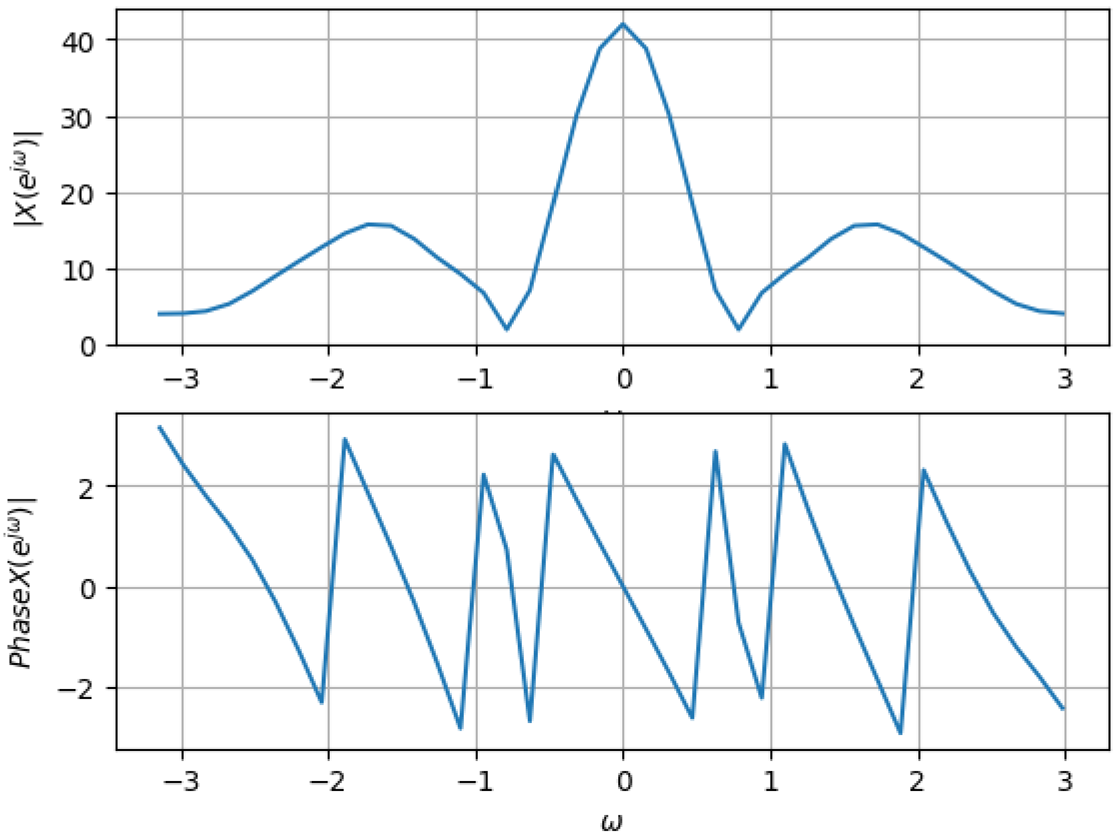
In [4]:
```python
def magnitude(inp):
    N = inp.__len__()
    output = [0]*N
    for k in range(N):
        output[k]= math.sqrt(inp[k].real**2 + inp[k].imag**2)
    return output
```

In [5]:
```python
def phase(inp):
    N = inp.__len__()
    output = [0]*N
    for k in range(N):
        output[k]= math.atan2(inp[k].imag,inp[k].real)
    return output
```

In [6]:
```python
Xf = dtft(x, freqPts)
Xf
```

Out[6]:
```
[(-4+2.326828918379971e-15j),
 (-3.033431192062262+2.6858506066380174j),
 (-0.9404349444493789+4.257458801593723j),
 (1.8947074025488293+4.960617334359528j),
 (6.04508497187474+3.493955364313913j),
 (8.535533905932734-2.707106781186555j),
 (3.2835027782176227-10.368745754912698j),
 (-8.524668919934165-9.4802211078134j),
 (-14.16311896062463+3.3021975254650515j),
 (-4.339120277515358+15.118976652643312j),
 (11.000000000000005+10.999999999999995j),
 (13.10968651765803-4.405757208594618j),
 (0.45491502812525964-11.359677632405859j),
 (-8.752405403728892-2.938515774500834j),
 (-4.047434800717829+5.40318760866132j),
 (1.4644660940672614+1.2928932188134512j),
 (-6.3368810393753705-3.216440812888141j),
 (-15.922584763882824+9.363183911046118j),
 (-4.295633033050406+29.776983052154915j),
 (25.567816636916668+29.154856377875085j),
 (42+0j),
 (25.567816636916668-29.154856377875085j),
 (-4.295633033050406-29.776983052154915j),
 (-15.922584763882824-9.363183911046118j),
 (-6.3368810393753705+3.216440812888141j),
 (1.4644660940672614-1.2928932188134512j),
 (-4.047434800717829-5.40318760866132j),
 (-8.752405403728892+2.938515774500834j),
 (0.45491502812525964+11.359677632405859j),
 (13.10968651765803+4.405757208594618j),
 (11.000000000000005-10.999999999999995j),
 (-4.339120277515358-15.118976652643312j),
 (-14.16311896062463-3.3021975254650515j),
 (-8.524668919934165+9.4802211078134j),
 (3.2835027782176227+10.368745754912698j),
 (8.535533905932734+2.707106781186555j),
 (6.04508497187474-3.493955364313913j),
 (1.8947074025488293-4.960617334359528j),
 (-0.9404349444493789-4.257458801593723j),
 (-3.033431192062262-2.6858506066380174j)]
```

In [7]:
```
plt.figure(1)
plt.subplot(2,1,1)
plt.plot(freqPts,magnitude(Xf))
plt.xlabel('$\omega$')
plt.ylabel('$|X(e^{j\omega})|$')
plt.grid()

plt.figure(1)
plt.subplot(2,1,2)
plt.plot(freqPts,phase(Xf))
plt.xlabel('$\omega$')
plt.ylabel('$Phase X(e^{j\omega})|$')
plt.grid()
```



Question 1 Part b

In [8]:
```
n = 10   #Time points
nf = 40 #Frequency points

w_start = 0
w_end = math.pi

x = [1, 2, 3, 4, 5, 10, 1, 2, 9, 5]
print(x)
freqPts = np.arange(w_start, w_end, (w_end - w_start)/nf)
freqPts
```

```
[1, 2, 3, 4, 5, 10, 1, 2, 9, 5]
```

Out[8]:
```
array([0.        , 0.07853982, 0.15707963, 0.23561945, 0.31415927,
       0.39269908, 0.4712389 , 0.54977871, 0.62831853, 0.70685835,
       0.78539816, 0.86393798, 0.9424778 , 1.02101761, 1.09955743,
       1.17809725, 1.25663706, 1.33517688, 1.41371669, 1.49225651,
       1.57079633, 1.64933614, 1.72787596, 1.80641578, 1.88495559,
       1.96349541, 2.04203522, 2.12057504, 2.19911486, 2.27765467,
       2.35619449, 2.43473431, 2.51327412, 2.59181394, 2.67035376,
       2.74889357, 2.82743339, 2.9059732 , 2.98451302, 3.06305284])
```

In [9]:
```
def dtft1(x,freqPts):
    output = [0]*freqPts.__len__()
    for k in range(freqPts.__len__()):
      W =  cm.exp(-1j * freqPts[k])
      W2 = cm.exp(-1j * freqPts[k] * 2)
```

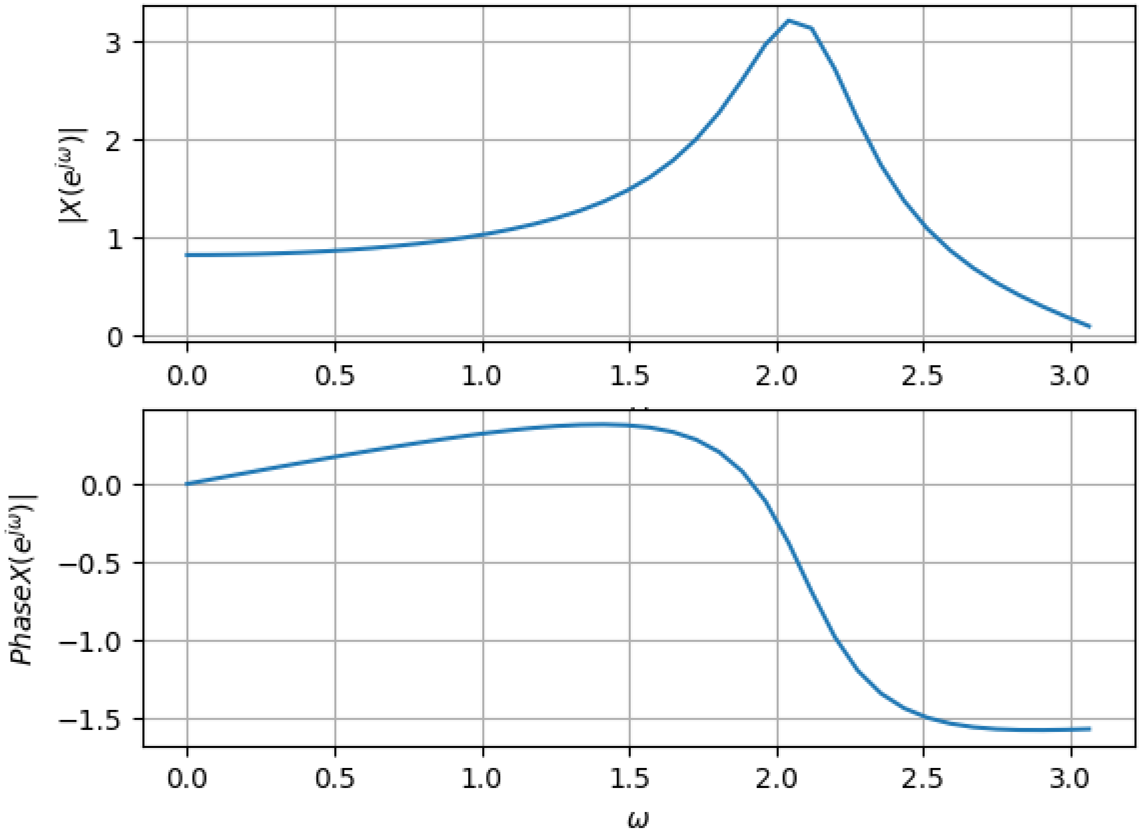
```
        output[k] = (1 + W)/(1 + 0.8*W + 0.64*W2)
    return output
```

In [10]:
```
Xf = dtft1(x, freqPts)
Xf
```

Out[10]:
```
[(0.819672131147541+0j),
 (0.8203713580761675+0.022705849438625123j),
 (0.8224827497563217+0.04550573906413985j),
 (0.8260480530919296+0.06849497855161353j),
 (0.8311389426707647+0.09177141746747296j),
 (0.8378603429739131+0.11543671002856862j),
 (0.8463554574731249+0.139597550342993j),
 (0.8568129399782047+0.16436682157201518j),
 (0.8694768613939406+0.18986454378454704j),
 (0.8846604358400647+0.21621840110013094j),
 (0.9027649266050223+0.2435634431411152j),
 (0.924305837851175+0.27204022201455047j),
 (0.9499495456699582+0.3017900172406483j),
 (0.9805651473011655+0.3329446712858156j),
 (1.01729885738995+0.36560641843525593j),
 (1.0616823051533373+0.39980893976210874j),
 (1.11579240936766+0.43544261313313337j),
 (1.1824901665673984+0.4721099887656411j),
 (1.2657791944005412+0.5088417303578443j),
 (1.3713386278946333+0.5435253629269168j),
 (1.507276507276507+0.5717255717255717j),
 (1.6850201311663093+0.5841862545679556j),
 (1.9196149002868046+0.5614803662931352j),
 (2.226089626810445+0.4629518650426246j),
 (2.5995755710280557+0.2080846155387476j),
 (2.947828191616794-0.3280653529857911j),
 (2.9829111641841553-1.186518404735528j),
 (2.4067246666081035-2.001751049252622j),
 (1.5041938170094147-2.2621902188236205j),
 (0.7935194132652268-2.049014483397459j),
 (0.38454163975584255-1.6938962493909877j),
 (0.17572525899254257-1.3607913580744886j),
 (0.07375364466710095-1.0861654460286985j),
 (0.02570521522485746-0.8651806526247199j),
 (0.004583636729761638-0.6854304518026705j),
 (-0.0032383111645213707-0.5357238353231543j),
 (-0.004761141102149136-0.40744153601505556j),
 (-0.0036618667097315463-0.2941449974332166j),
 (-0.0019005778034483873-0.19094877214069578j),
 (-0.0005125962946409549-0.09398658351734916j)]
```

In [11]:
```
plt.figure(1)
plt.subplot(2,1,1)
plt.plot(freqPts,magnitude(Xf))
plt.xlabel('$\omega$')
plt.ylabel('$|X(e^{j\omega})|$')
plt.grid()

plt.figure(1)
plt.subplot(2,1,2)
```

```
plt.plot(freqPts,phase(Xf))
plt.xlabel('$\omega$')
plt.ylabel('$Phase X(e^{j\omega})|$')
plt.grid()
```



Question 1 Part c

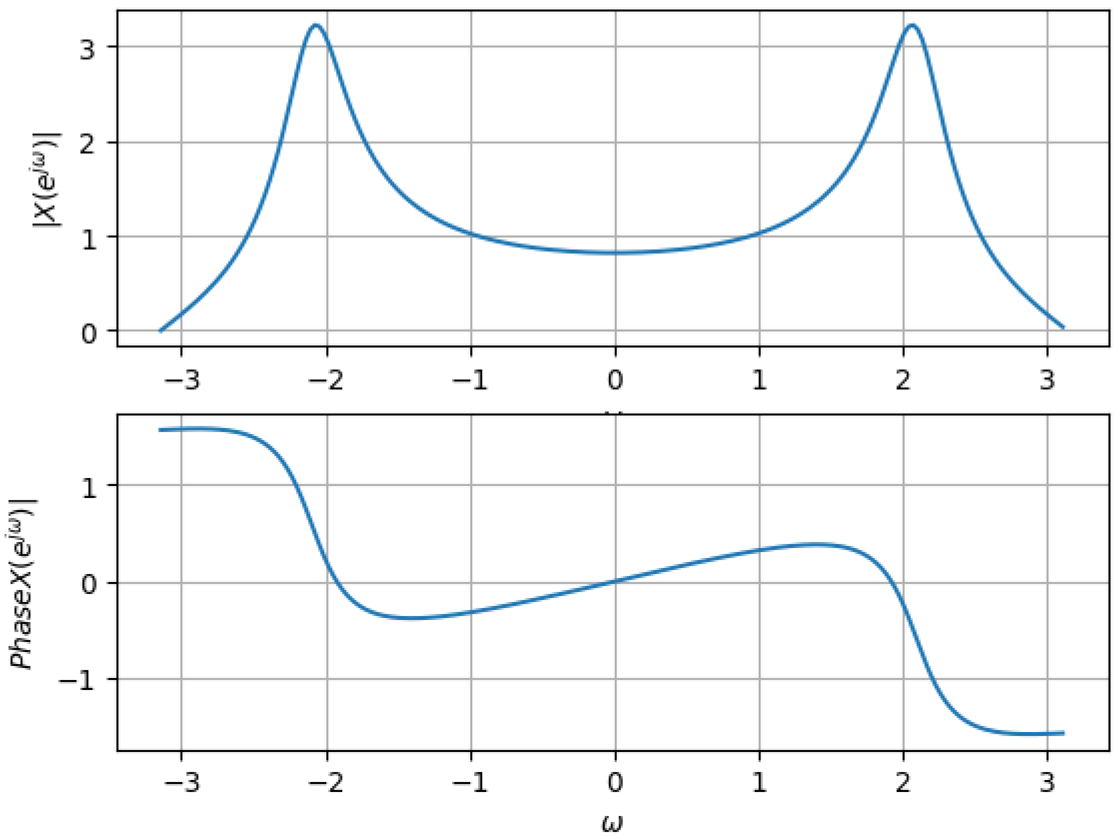
In [12]:
```
nf = 200 #Frequency points

w_start = -math.pi
w_end = math.pi

n = 15   #Time points
x = [0]*n
for t in range(n):
  x[t] = math.cos(0.4 * math.pi * t)

print(x)

freqPts = np.arange(w_start, w_end, (w_end - w_start)/nf)
```

```
[1.0, 0.30901699437494745, -0.8090169943749473, -0.8090169943749475, 0.30901699437494723, 1.0, 0.30901699437494773, -0.8090169943749472, -0.8090169943749477, 0.309016994374947, 1.0, 0.30901699437494795, -0.8090169943749471, -0.8090169943749478, 0.3090169943749468]
```

In [13]:
```
Xf = dtft1(x,freqPts)
```

In [14]:
```
plt.figure(1)
plt.subplot(2,1,1)
```

```
plt.plot(freqPts,magnitude(Xf))
plt.xlabel('$\omega$')
plt.ylabel('$|X(e^{j\omega})|$')
plt.grid()

plt.figure(1)
plt.subplot(2,1,2)
plt.plot(freqPts,phase(Xf))
plt.xlabel('$\omega$')
plt.ylabel('$Phase X(e^{j\omega})|$')
plt.grid()
```



In [15]:
```
n = 63  #Time points
x = [0]*n
for t in range(n):
  x[t] = math.cos(0.4 * math.pi * t)
print(x)
```

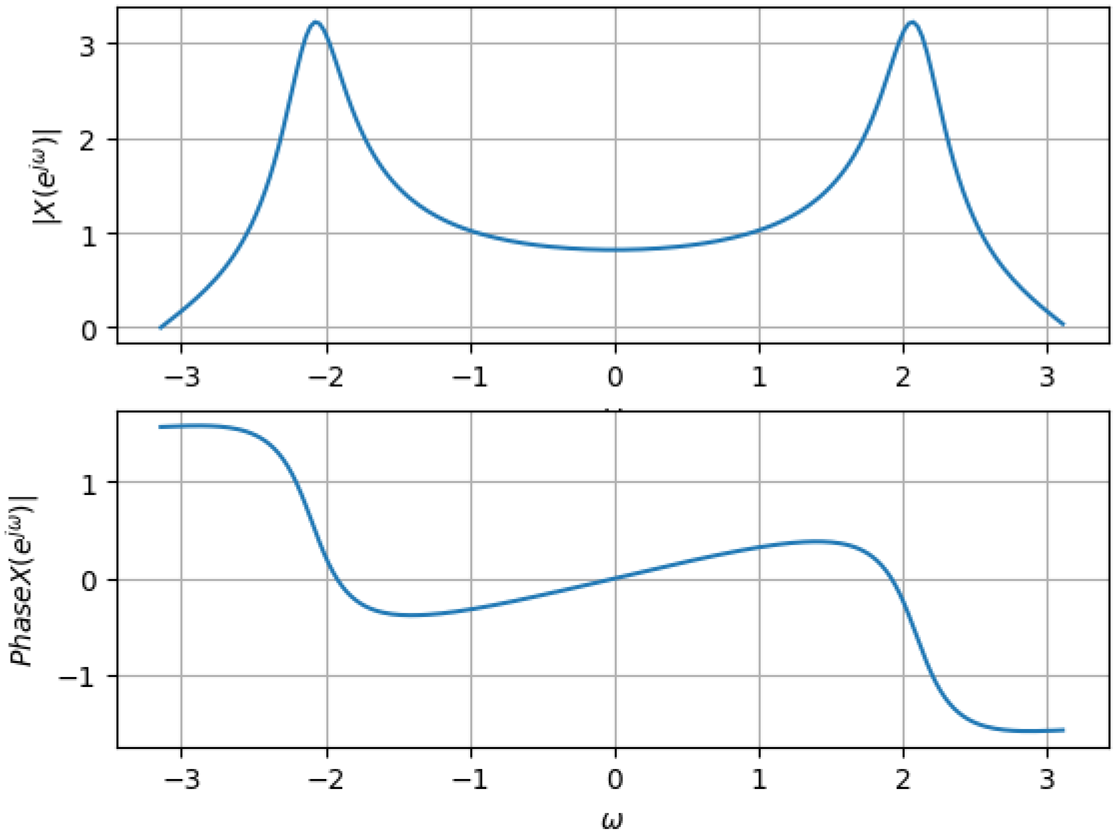
```
[1.0, 0.30901699437494745, -0.8090169943749473, -0.8090169943749475, 0.30901699437
494723, 1.0, 0.30901699437494773, -0.8090169943749472, -0.8090169943749477, 0.3090
16994374947, 1.0, 0.30901699437494795, -0.8090169943749471, -0.8090169943749478,
0.3090169943749468, 1.0, 0.3090169943749482, -0.8090169943749469, -0.8090169943749
479, 0.30901699437494656, 1.0, 0.3090169943749484, -0.8090169943749468, -0.8090169
943749481, 0.3090169943749463, 1.0, 0.3090169943749486, -0.8090169943749467, -0.80
90169943749482, 0.30901699437494606, 1.0, 0.3090169943749489, -0.8090169943749466,
-0.8090169943749483, 0.30901699437494584, 1.0, 0.3090169943749491, -0.809016994374
9463, -0.8090169943749485, 0.3090169943749456, 1.0, 0.30901699437494934, -0.809016
9943749462, -0.8090169943749487, 0.3090169943749454, 1.0, 0.30901699437494956, -0.
8090169943749461, -0.8090169943749488, 0.3090169943749451, 1.0, 0.3090169943749565
6, -0.8090169943749459, -0.8090169943749448, 0.3090169943749449, 1.0, 0.3090169943
7495, -0.80901699437495, -0.8090169943749491, 0.3090169943749379, 1.0, 0.309016994
3749435, -0.8090169943749457]
```

In [16]:

```python
Xf = dtft1(x,freqPts)
```

In [17]:

```python
plt.figure(1)
plt.subplot(2,1,1)
plt.plot(freqPts,magnitude(Xf))
plt.xlabel('$\omega$')
plt.ylabel('$|X(e^{j\omega})|$')
plt.grid()

plt.figure(1)
plt.subplot(2,1,2)
plt.plot(freqPts,phase(Xf))
plt.xlabel('$\omega$')
plt.ylabel('$Phase X(e^{j\omega})|$')
plt.grid()
```

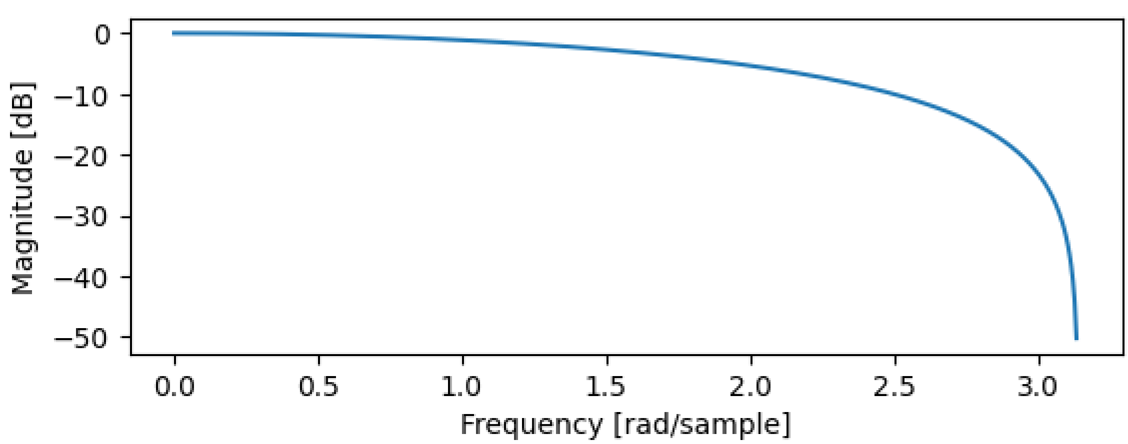


Question 1 Part d

In [18]:
```
L=2 #L-point filter
b = (np.ones(L))/L #numerator co-effs of filter transfer function
a = np.ones(1)

w, h = freqz(b,a)
```

In [19]:
```
plt.subplot(2, 1, 1)
plt.plot(w, 20 * np.log10(abs(h)))
plt.ylabel('Magnitude [dB]')
plt.xlabel('Frequency [rad/sample]')
```

Out[19]:
```
Text(0.5, 0, 'Frequency [rad/sample]')
```



Question 2 Part 1

In [20]:
```
def dft(x):
    #Linear transformation using W matrix for an N point DFT
    N2 = x.__len__()
    x = np.array(x)
    #the twiddle factor
    W=[[0 + 0j]*N2]*N2
    W = np.array(W)
    #creating the W matrix
    for k in range(N2):
     for n in range(N2):
      m = cm.exp(-1j * 2 * math.pi / N2 * k * n)
      W[k][n] = m

    X=np.matmul(W, np.transpose(x))
    return X
```

In [21]:
```
dft([1, 2, 3, 4, 3, 5])
```

Out[21]:
```
array([18. +0.00000000e+00j, -2.5+2.59807621e+00j, -1.5+2.59807621e+00j,
       -4. -2.57175828e-15j, -1.5-2.59807621e+00j, -2.5-2.59807621e+00j])
```

In [22]:
```
np.fft.fft(np.array([1, 2, 3, 4, 3, 5]))
```

Out[22]:
```
array([18. +0.00000000e+00j, -2.5+2.59807621e+00j, -1.5+2.59807621e+00j,
       -4. -2.22044605e-16j, -1.5-2.59807621e+00j, -2.5-2.59807621e+00j])
```

Question 2 Part 2

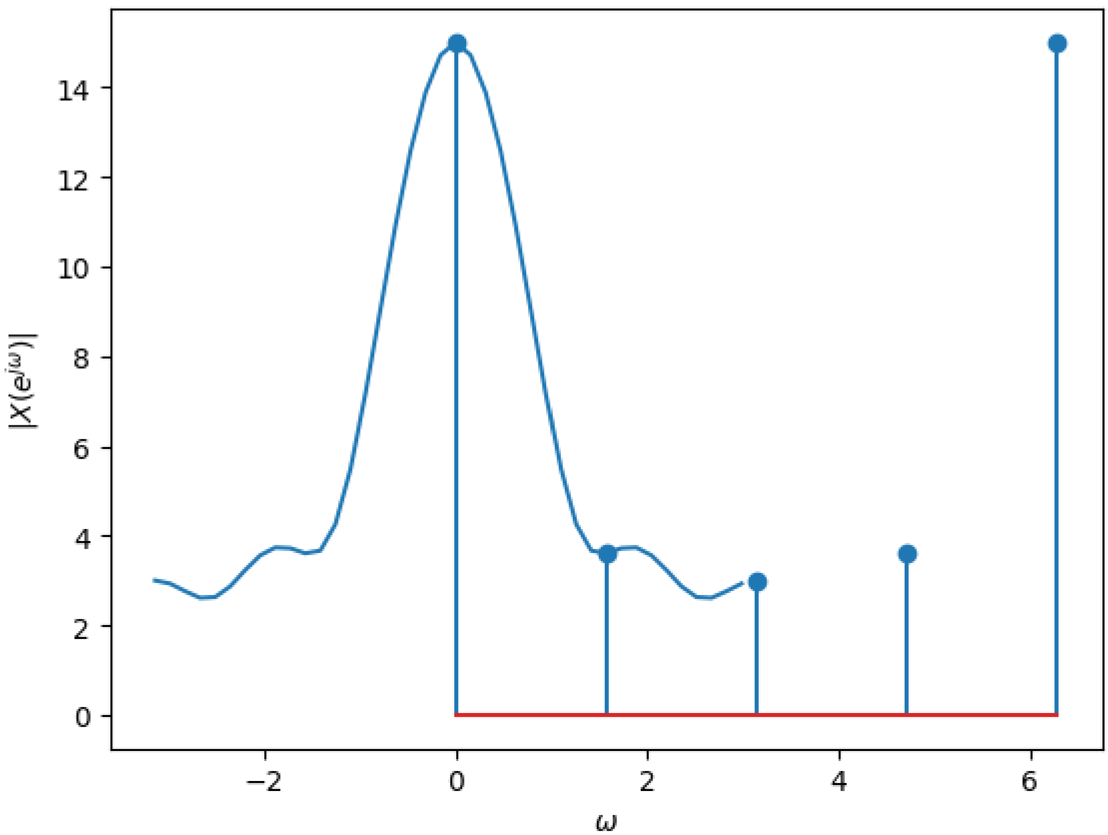
In [23]:
```
nf = 40 #Frequency points

w_start = -math.pi
w_end = math.pi

x=[1, 2, 3, 4, 5]
n = x.__len__()

print(x)
freqPts = np.arange(w_start, w_end, (w_end - w_start)/nf)
freqPts
```

```
[1, 2, 3, 4, 5]
```

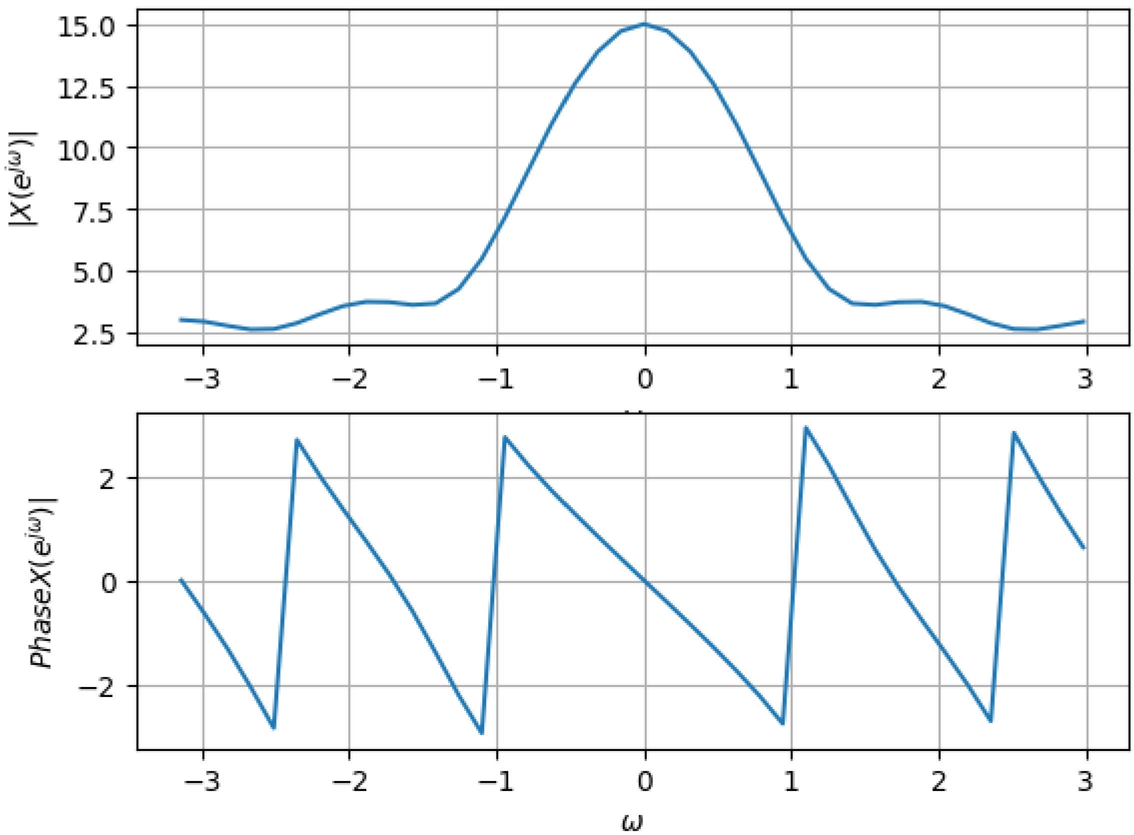
Out[23]:
```
array([-3.14159265, -2.98451302, -2.82743339, -2.67035376, -2.51327412,
       -2.35619449, -2.19911486, -2.04203522, -1.88495559, -1.72787596,
       -1.57079633, -1.41371669, -1.25663706, -1.09955743, -0.9424778 ,
       -0.78539816, -0.62831853, -0.4712389 , -0.31415927, -0.15707963,
        0.        ,  0.15707963,  0.31415927,  0.4712389 ,  0.62831853,
        0.78539816,  0.9424778 ,  1.09955743,  1.25663706,  1.41371669,
        1.57079633,  1.72787596,  1.88495559,  2.04203522,  2.19911486,
        2.35619449,  2.51327412,  2.67035376,  2.82743339,  2.98451302])
```

In [24]:
```
w = np.linspace(0, 2 * w_end, n)
X1 = dtft(x, w)
X2 = dtft(x, freqPts)
```

In [25]:
```
plt.figure(1)

# plt.subplot(2,1,1)
plt.stem(w, magnitude(X1))
plt.xlabel('$\omega$')
plt.ylabel('$|X(e^{j\omega})|$')

# plt.subplot(2,1,2)
plt.plot(freqPts,magnitude(X2))
plt.xlabel('$\omega$')
plt.ylabel('$|X(e^{j\omega})|$')

plt.show()
```



In [26]:
```
plt.figure(1)
plt.subplot(2,1,1)
plt.plot(freqPts,magnitude(X2))
plt.xlabel('$\omega$')
plt.ylabel('$|X(e^{j\omega})|$')
plt.grid()

plt.figure(1)
plt.subplot(2,1,2)
plt.plot(freqPts,phase(X2))
plt.xlabel('$\omega$')
plt.ylabel('$Phase X(e^{j\omega})|$')
plt.grid()
```



15.0
12.5
10.0
7.5
5.0
2.5
$|X(e^{j\omega})|$
−3
−2
−1
0
1
2
3
$Phase X(e^{j\omega})|$
2
0
−2
ω

In [ ]: