In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy
from scipy import signal
```

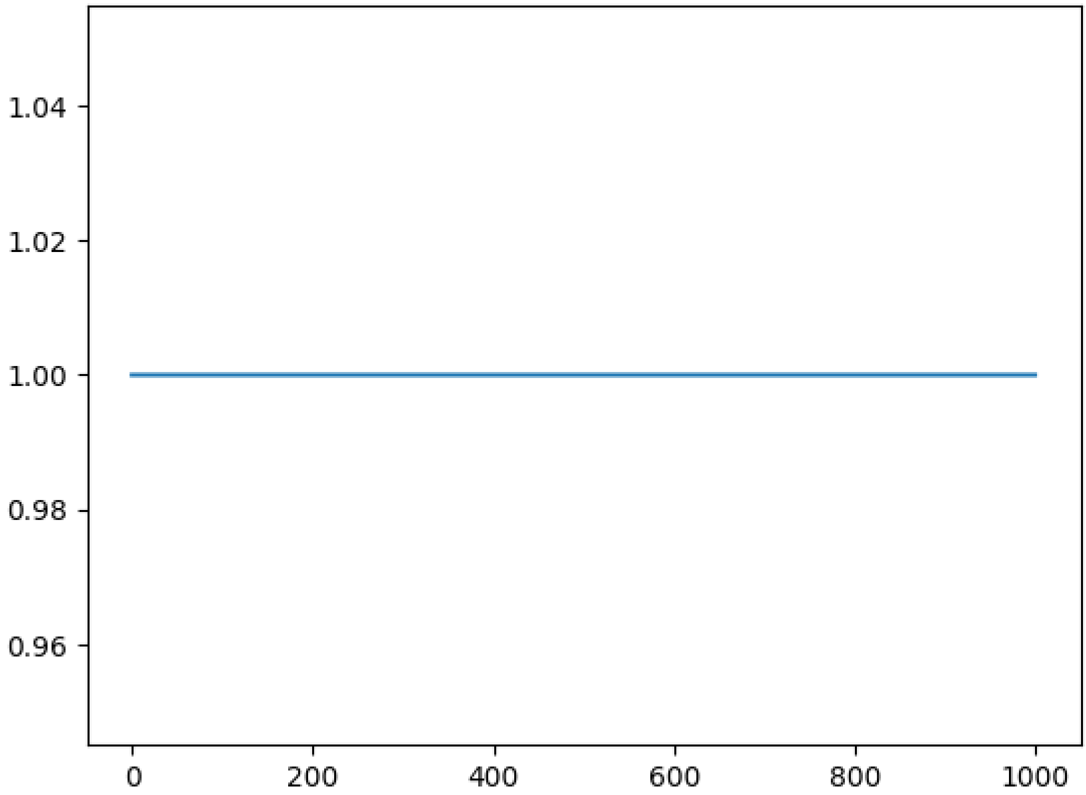
Question 2:A

In [ ]:
```
freq = np.arange(0, 1000, 0.1)

# H(S) = 1

H = np.ones(len(freq))
plt.plot(freq, H)
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x21ec5080ec8>]
```



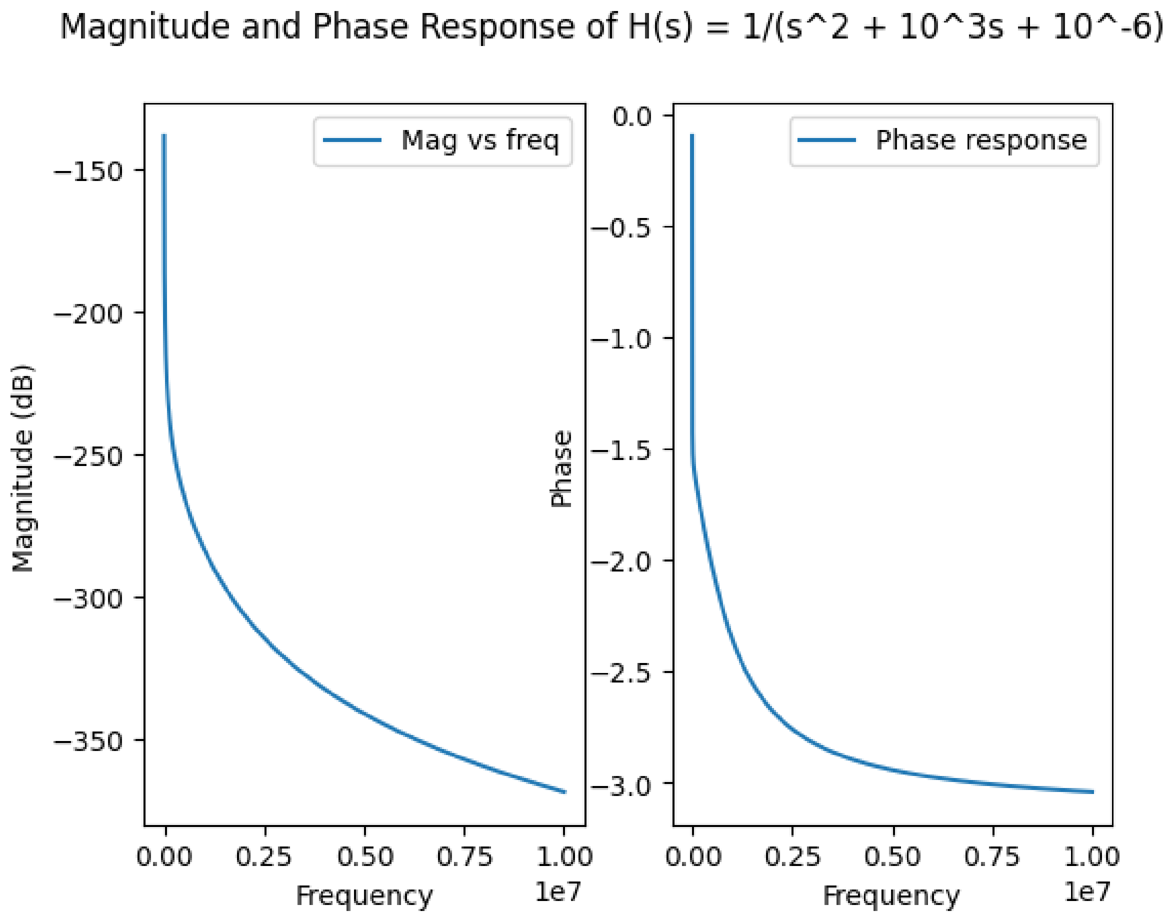
Question 2:B

H(S) = S/(S + 1000 + S^2 * 10^-6), R out = 1000 Ohm, C1 = 10^-6 F, L1 = 1 mH,

In [ ]:
```
num = [1]
den = [10**-6, 1, 10**3]
tf = signal.TransferFunction(num, den) # create transfer function
zpg = tf.to_zpk() # convert to zero-pole-gain form
w, H = signal.freqresp(zpg) # calculate frequency response
```

In [ ]:
```
# create subplot 1
fig, ax = plt.subplots(1,2)
ax[0].plot(w, 20*np.log(abs(H)), label="Mag vs freq")
ax[0].legend()
ax[0].set_xlabel('Frequency')
ax[0].set_ylabel('Magnitude (dB)')
```

```
# size of subplot 2
# ax[0].set_figwidth(10)
ax[1].plot(w, np.angle(H), label="Phase response")
ax[1].legend()
ax[1].set_xlabel('Frequency')
ax[1].set_ylabel('Phase')
fig.suptitle('Magnitude and Phase Response of H(s) = 1/(s^2 + 10^3s + 10^-6)')
plt.show()
```



Question 2:C; Lout = 2k,

H(S) = S*Rout

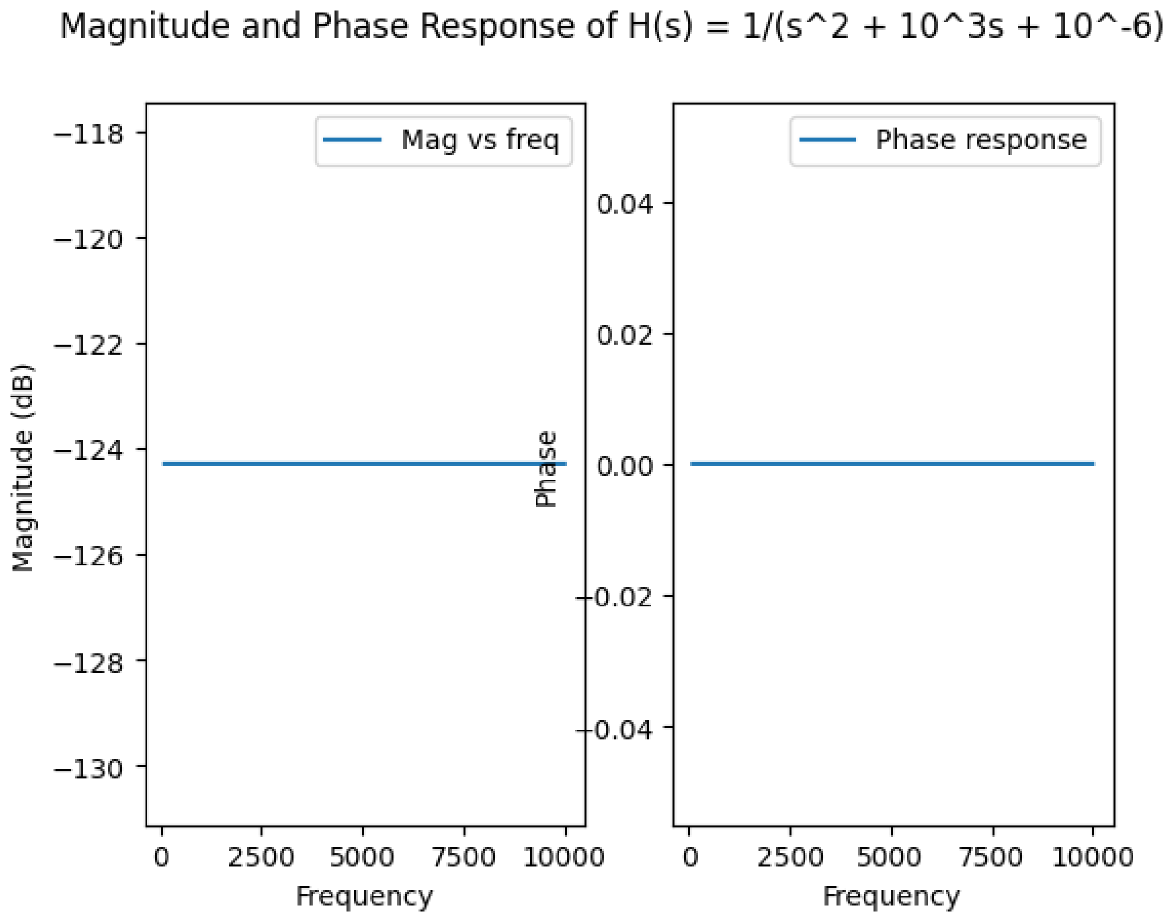
In [ ]:
```
num = [2*10**-3]
den = [1]

tf = signal.TransferFunction(num, den) # create transfer function
zpg = tf.to_zpk() # convert to zero-pole-gain form
w, H = signal.freqresp(zpg) # calculate frequency response
tf
```

Out[ ]:
```
TransferFunctionContinuous(
array([0.002]),
array([1.]),
dt: None
)
```

In [ ]:
```
fig, ax = plt.subplots(1,2)
ax[0].plot(w, 20*np.log(abs(H)), label="Mag vs freq")
ax[0].legend()
ax[0].set_xlabel('Frequency')
ax[0].set_ylabel('Magnitude (dB)')
# size of subplot 2
```

```
ax[1].plot(w, np.angle(H), label="Phase response")
ax[1].legend()
ax[1].set_xlabel('Frequency')
ax[1].set_ylabel('Phase')
fig.suptitle('Magnitude and Phase Response of H(s) = 1/(s^2 + 10^3s + 10^-6)')
plt.show()
```



Question 2:D, L2 = 2mH, Rout = 1k, C1 = 1 micro F,

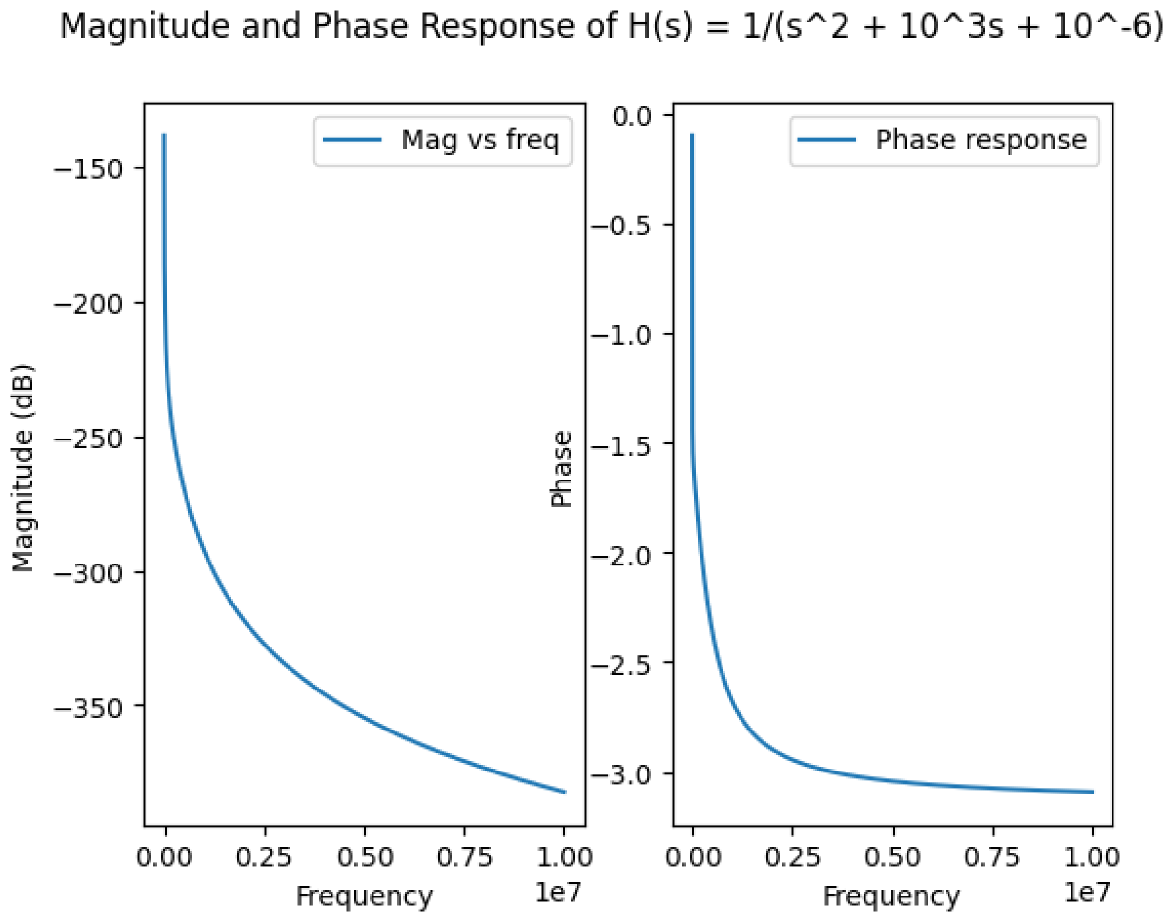
H(S) = S/ [S^2 * (L2/Rout) + S + 1/(C1*Rout)]

In [ ]:
```
num = [1]
den = [(2*10**-3)/10**3, 1, 1/(10**-6*10**3)]
tf = signal.TransferFunction(num, den) # create transfer function
zpg = tf.to_zpk() # convert to zero-pole-gain form
w, H = signal.freqresp(zpg) # calculate frequency response
tf
```

Out[ ]:
```
TransferFunctionContinuous(
array([500000.]),
array([1.e+00, 5.e+05, 5.e+08]),
dt: None
)
```

In [ ]:
```
fig, ax = plt.subplots(1,2)
ax[0].plot(w, 20*np.log(abs(H)), label="Mag vs freq")
ax[0].legend()
ax[0].set_xlabel('Frequency')
ax[0].set_ylabel('Magnitude (dB)')
# size of subplot 2
# ax[0].set_figwidth(10)
```

```
ax[1].plot(w, np.angle(H), label="Phase response")
ax[1].legend()
ax[1].set_xlabel('Frequency')
ax[1].set_ylabel('Phase')
fig.suptitle('Magnitude and Phase Response of H(s) = 1/(s^2 + 10^3s + 10^-6)')
plt.show()
```



In [ ]: