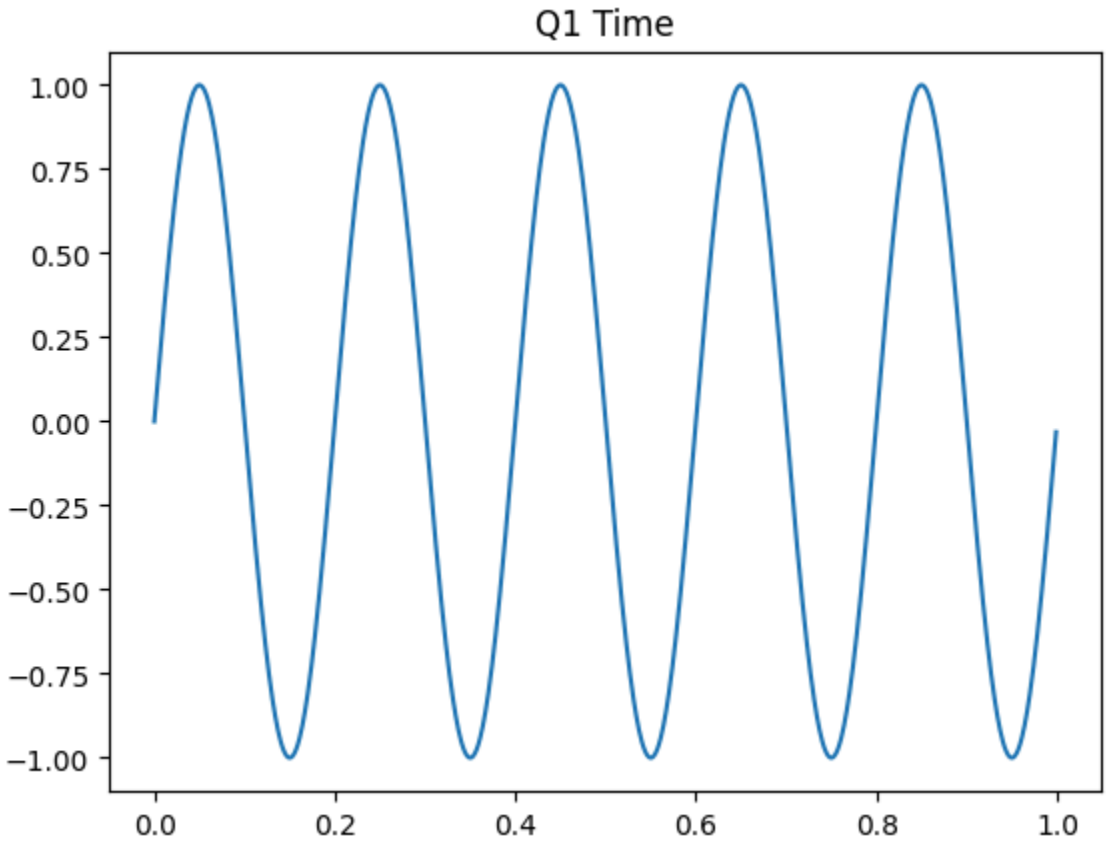
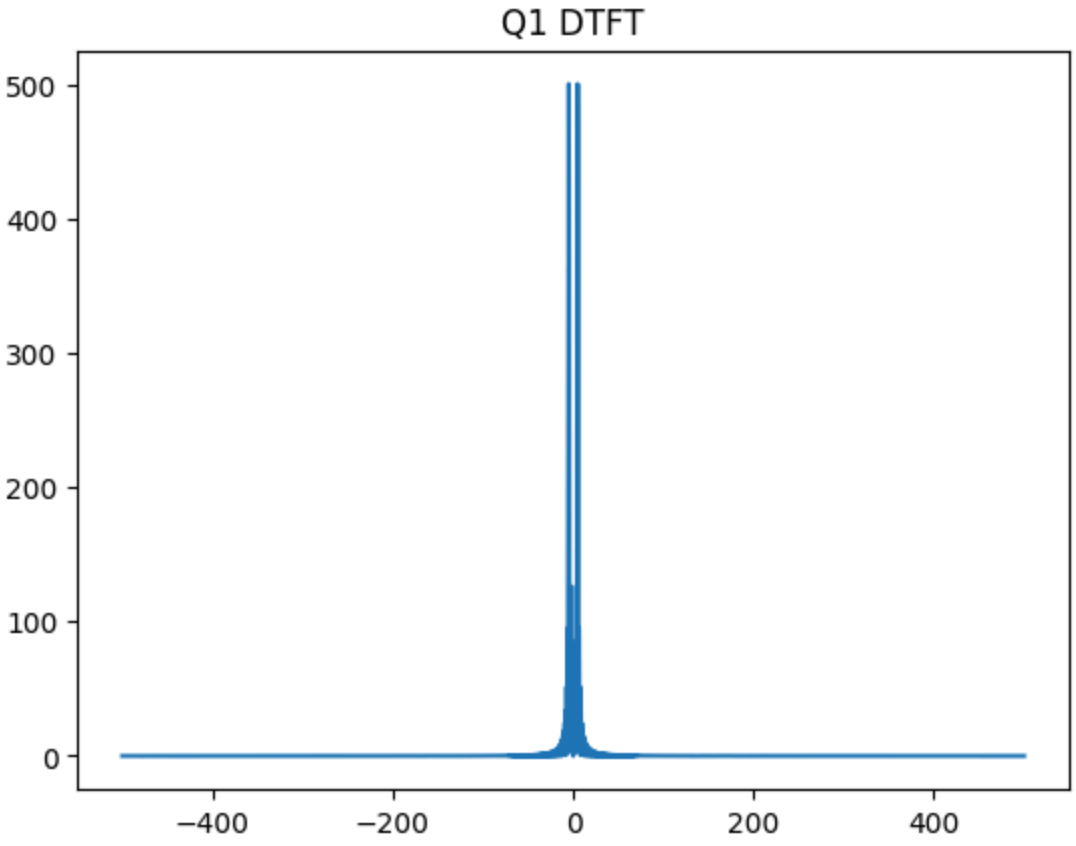
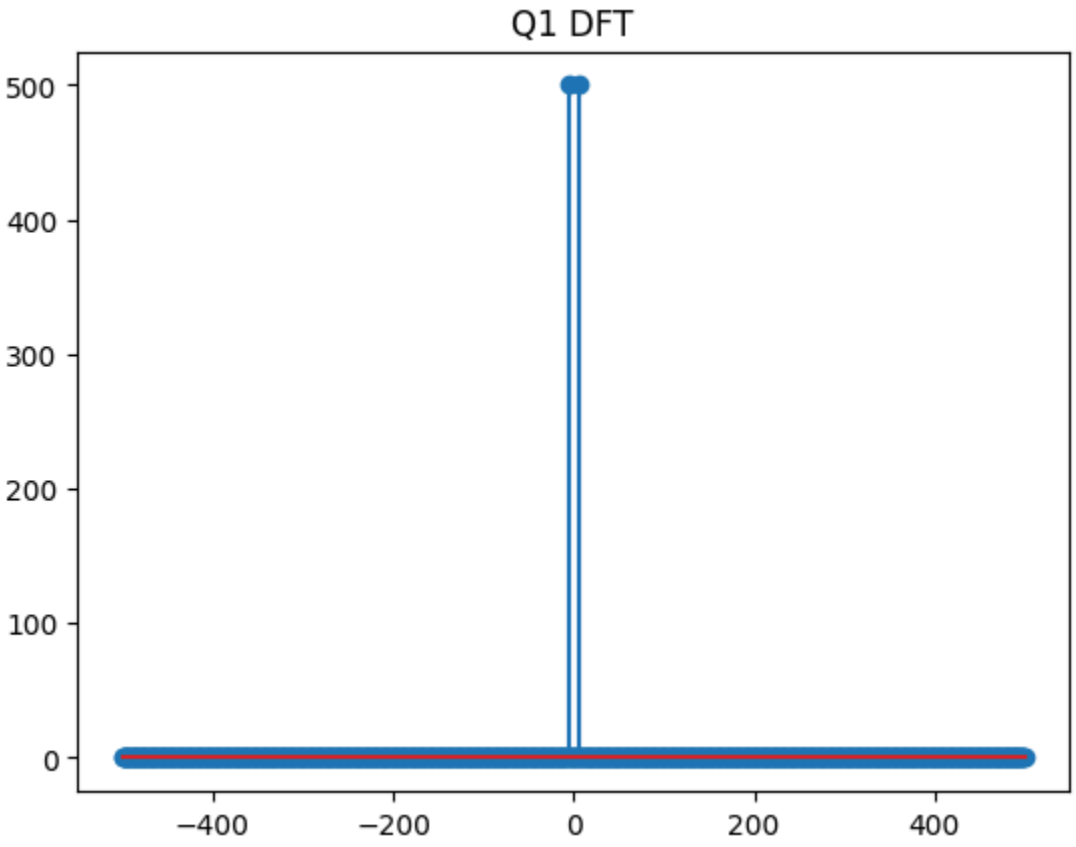
In [2]:
```
import numpy as np
import matplotlib.pyplot as plt

fs = 1000
t = np.arange(0,1,1/fs)
```

In [3]:
```
# Q1: Sinusoidal
x1 = np.sin(2*np.pi*5*t)
plt.plot(t,x1);plt.title("Q1 Time");plt.show()
X1_dtft = np.fft.fftshift(np.fft.fft(x1,16384))
freqs1 = np.fft.fftshift(np.fft.fftfreq(16384,1/fs))
plt.plot(freqs1,abs(X1_dtft));plt.title("Q1 DTFT");plt.show()
X1_dft = np.fft.fft(x1)
freqs1d = np.fft.fftfreq(len(x1),1/fs)
plt.stem(freqs1d,abs(X1_dft));plt.title("Q1 DFT");plt.show()
```

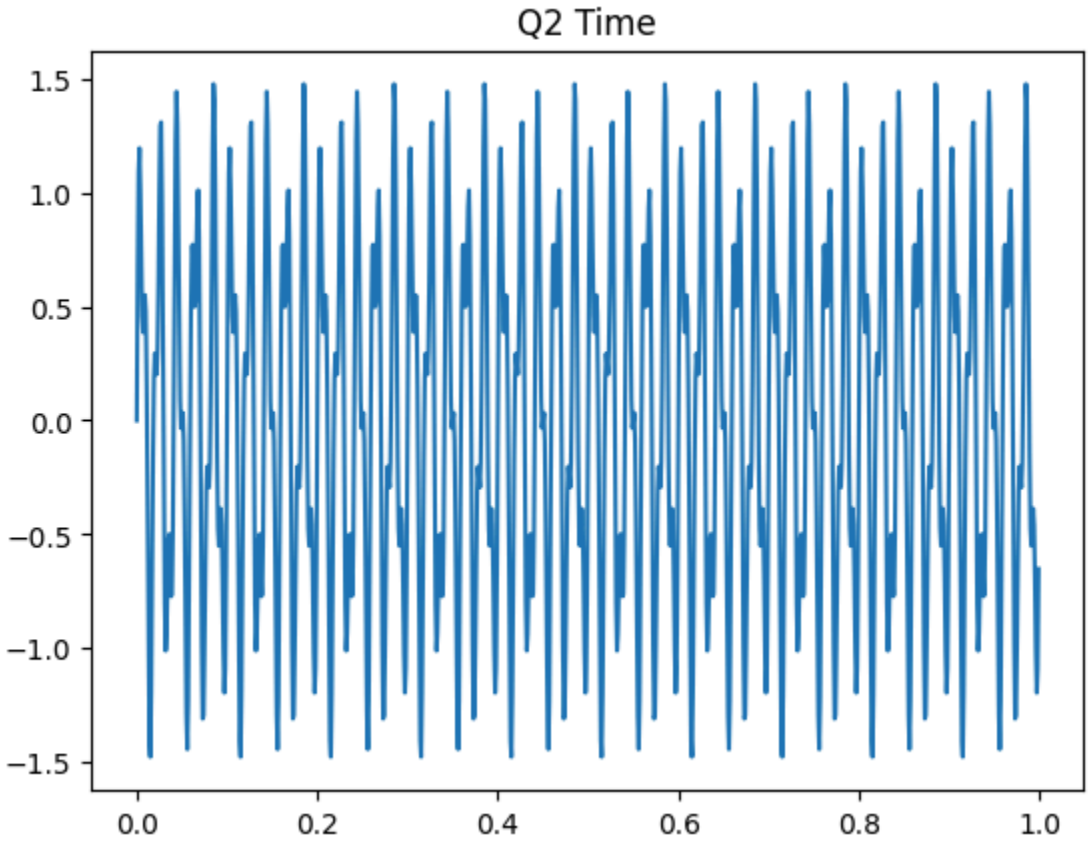


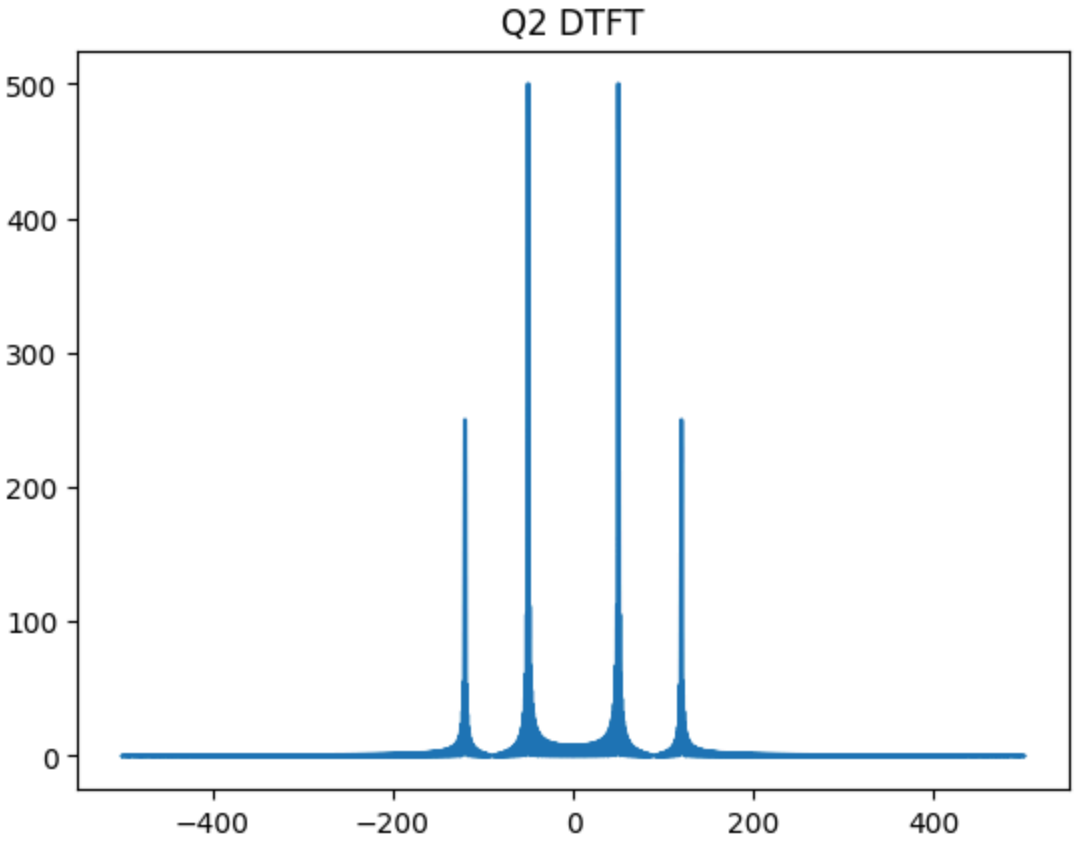


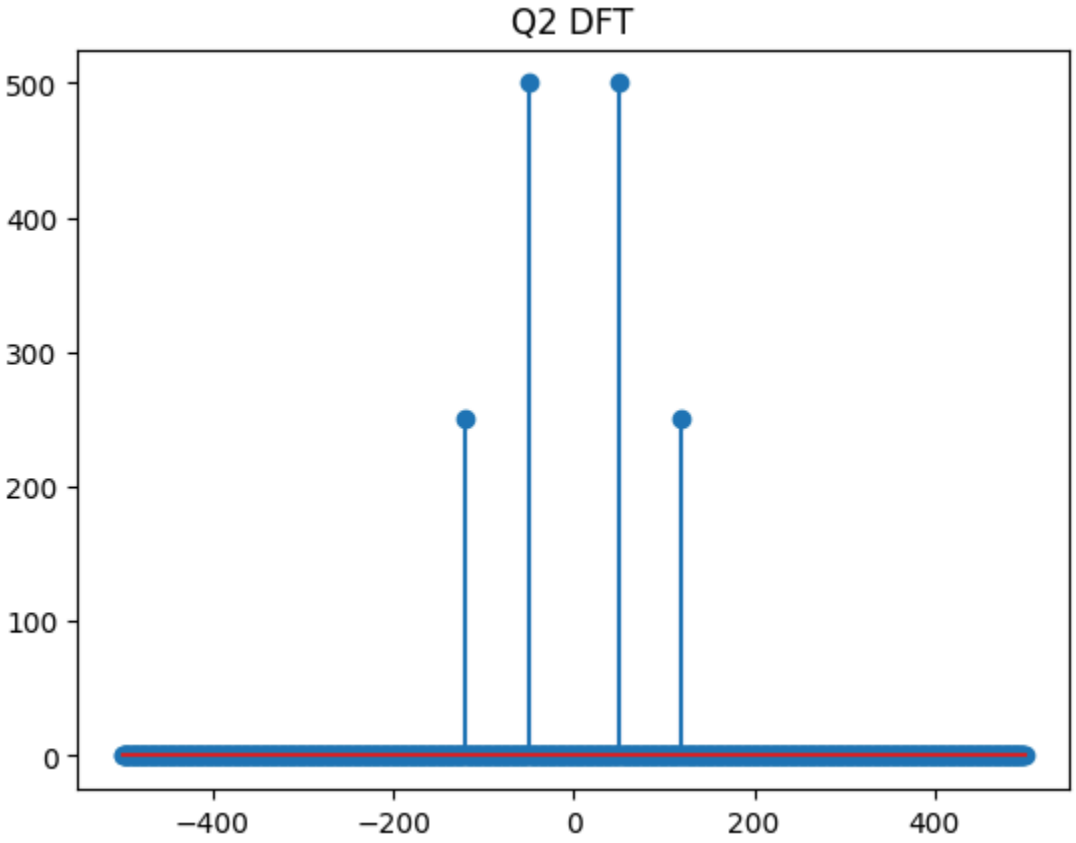


In [4]:
```
# Q2: Composite
```

```
x2 = np.sin(2*np.pi*50*t)+0.5*np.sin(2*np.pi*120*t)
plt.plot(t,x2);plt.title("Q2 Time");plt.show()
X2_dtft = np.fft.fftshift(np.fft.fft(x2,16384))
freqs2 = np.fft.fftshift(np.fft.fftfreq(16384,1/fs))
plt.plot(freqs2,abs(X2_dtft));plt.title("Q2 DTFT");plt.show()
X2_dft = np.fft.fft(x2)
freqs2d = np.fft.fftfreq(len(x2),1/fs)
plt.stem(freqs2d,abs(X2_dft));plt.title("Q2 DFT");plt.show()
```

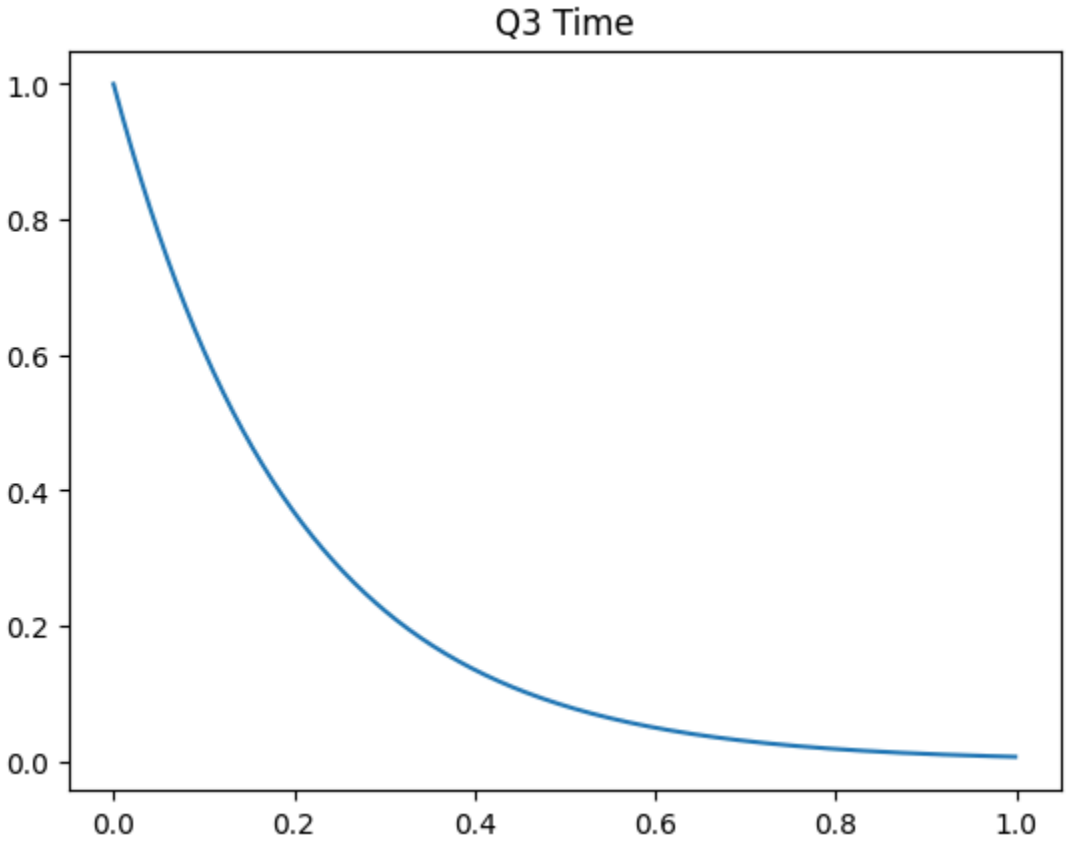


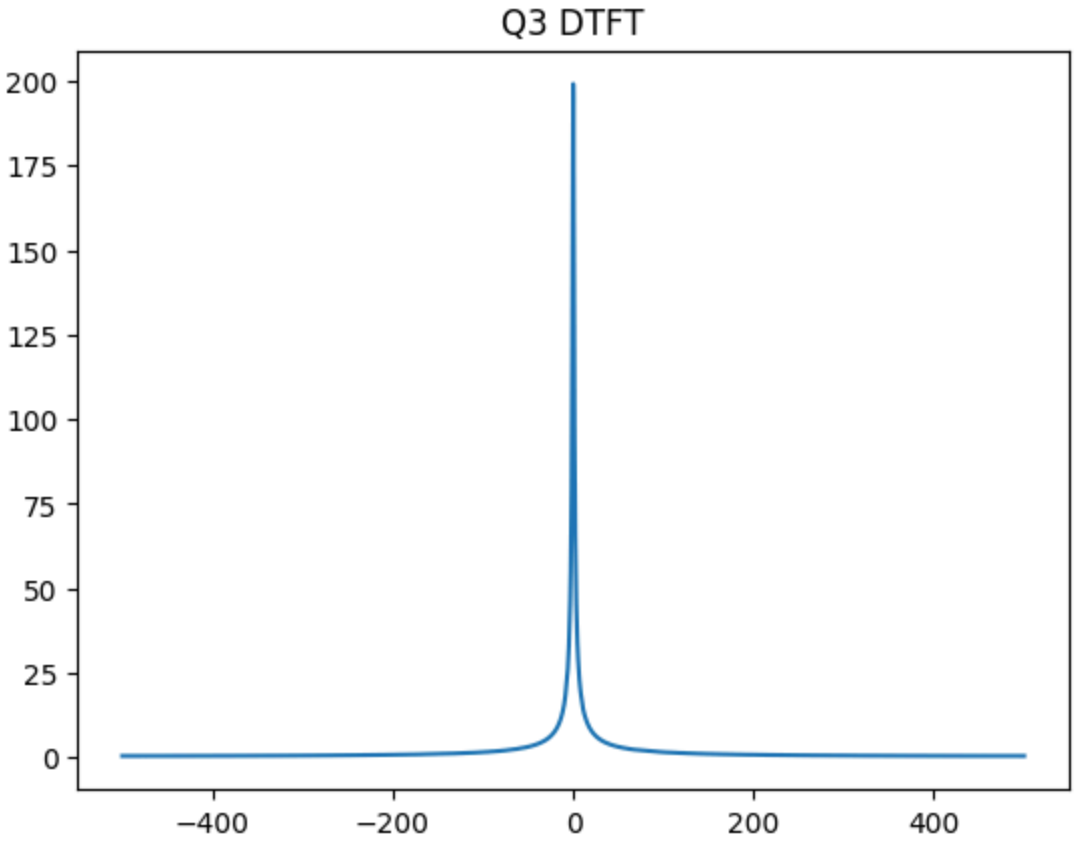


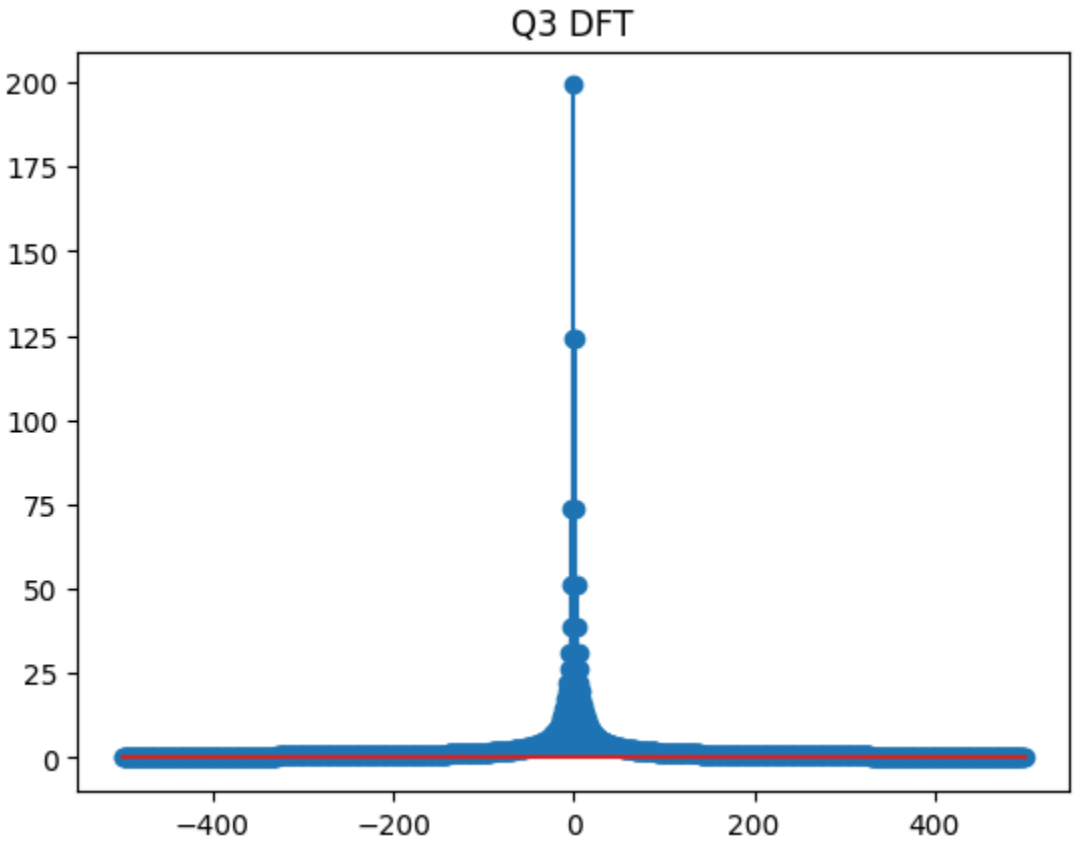


In [5]:
```
# Q3: Exponential
```

```
a = 5
x3 = np.exp(-a*t)
plt.plot(t,x3);plt.title("Q3 Time");plt.show()
X3_dtft = np.fft.fftshift(np.fft.fft(x3,16384))
freqs3 = np.fft.fftshift(np.fft.fftfreq(16384,1/fs))
plt.plot(freqs3,abs(X3_dtft));plt.title("Q3 DTFT");plt.show()
X3_dft = np.fft.fft(x3)
freqs3d = np.fft.fftfreq(len(x3),1/fs)
plt.stem(freqs3d,abs(X3_dft));plt.title("Q3 DFT");plt.show()
```

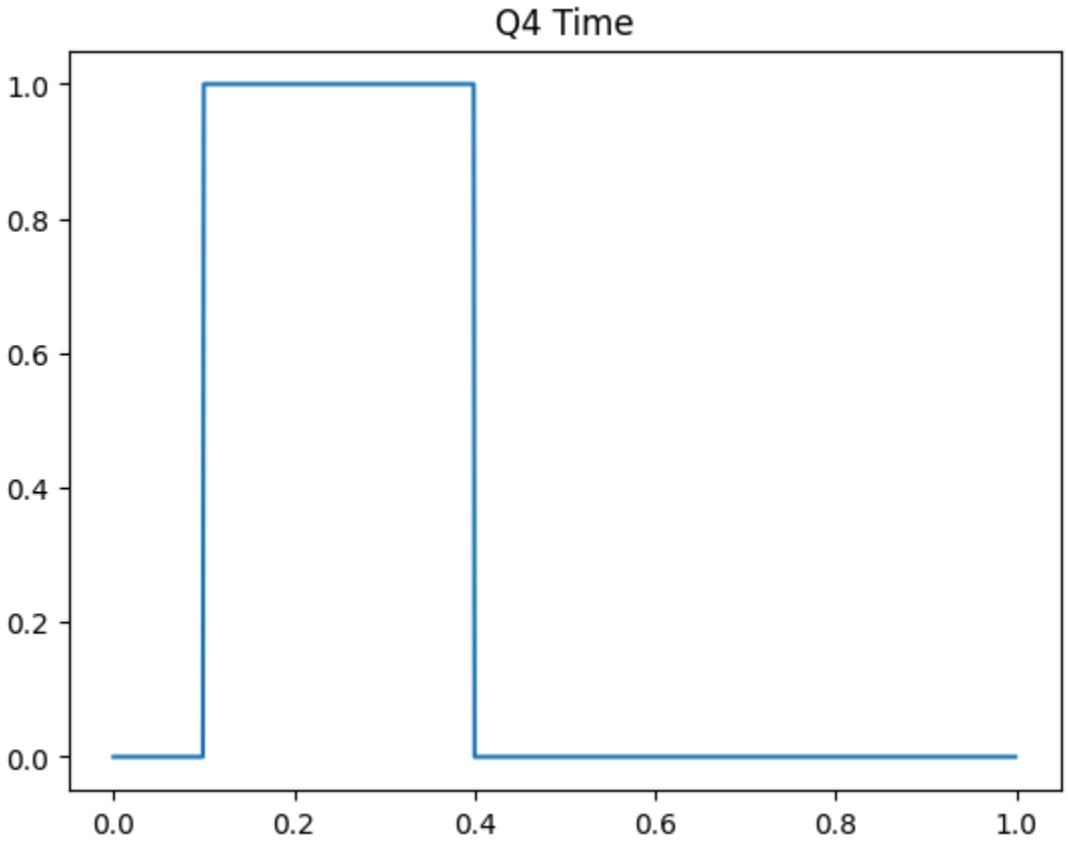


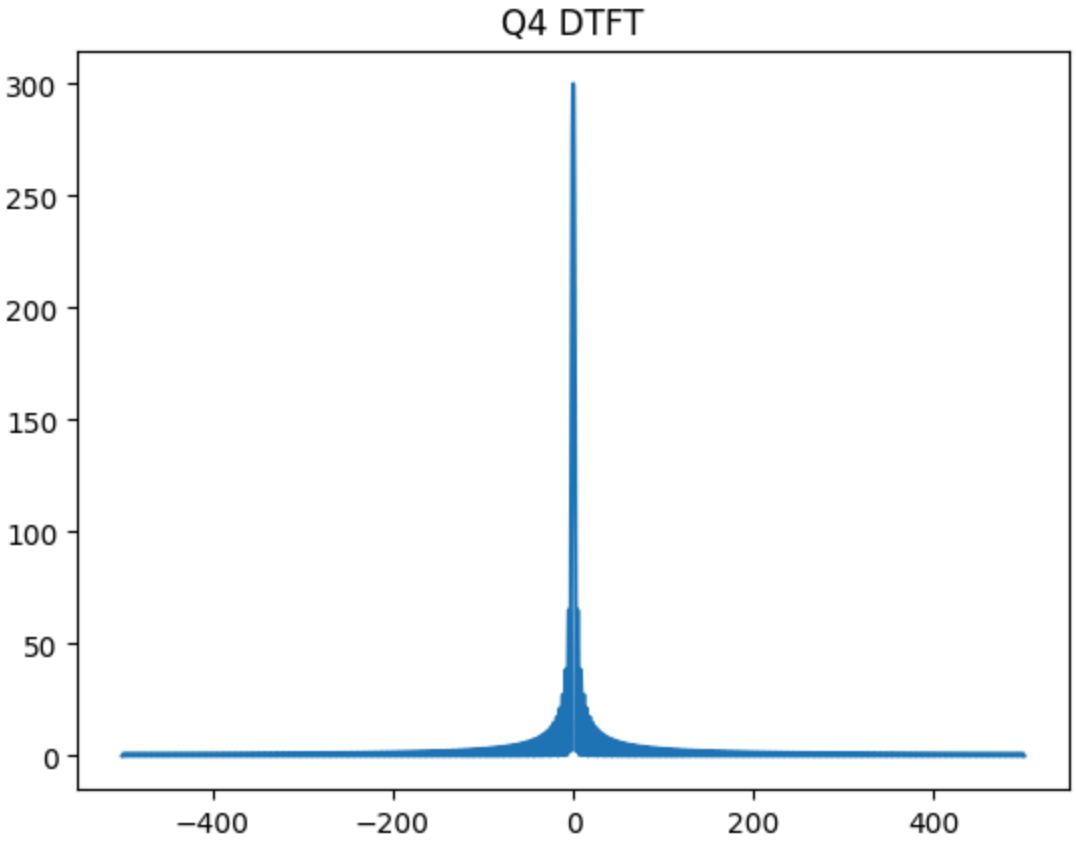


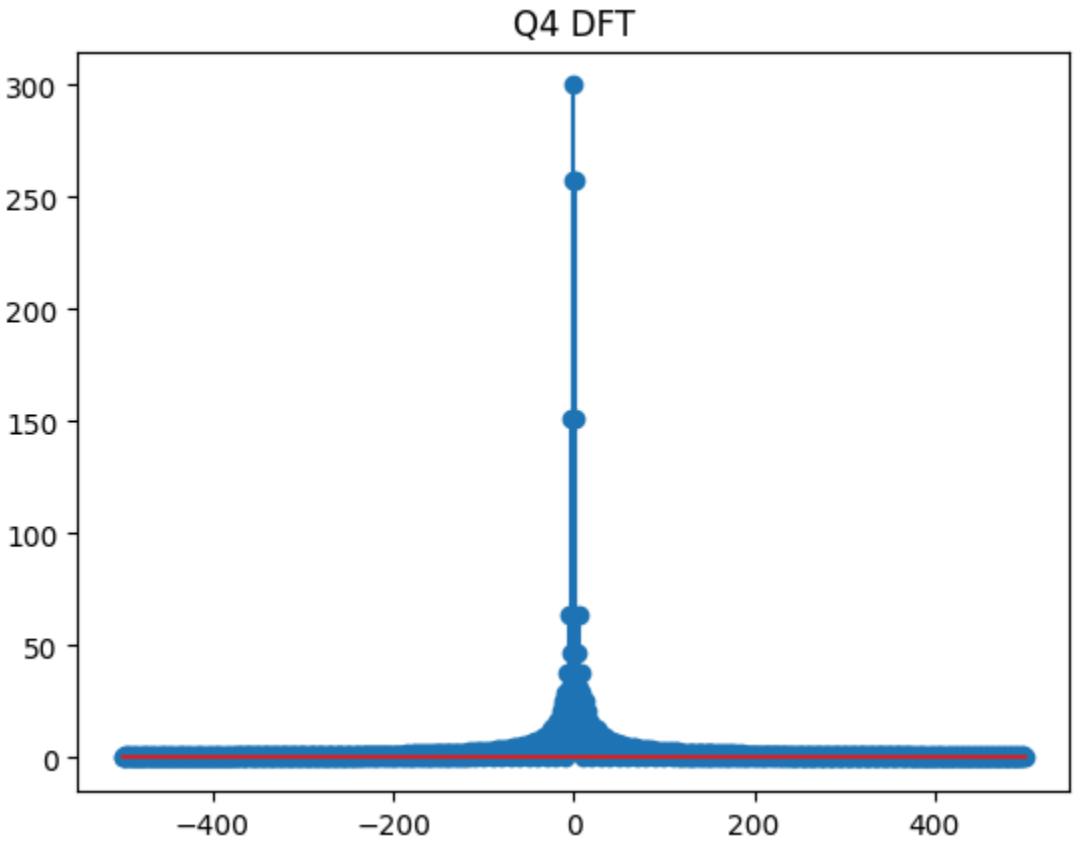


In [6]:
```
# Q4: Rectangular
```

```
x4 = np.zeros(len(t))
x4[100:400]=1
plt.plot(t,x4);plt.title("Q4 Time");plt.show()
X4_dtft = np.fft.fftshift(np.fft.fft(x4,16384))
freqs4 = np.fft.fftshift(np.fft.fftfreq(16384,1/fs))
plt.plot(freqs4,abs(X4_dtft));plt.title("Q4 DTFT");plt.show()
X4_dft = np.fft.fft(x4)
freqs4d = np.fft.fftfreq(len(x4),1/fs)
plt.stem(freqs4d,abs(X4_dft));plt.title("Q4 DFT");plt.show()
```







In [12]:
```
# Rectangular window (original signal)
```

```
X2_rect = np.fft.fftshift(np.fft.fft(x2, 16384))
freqs2 = np.fft.fftshift(np.fft.fftfreq(16384, 1/fs))

# Hamming windowed signal
x2_windowed = x2 * np.hamming(len(x2))
X2_ham = np.fft.fftshift(np.fft.fft(x2_windowed, 16384))

#Magnitude Spectrum
plt.figure(figsize=(10,5))
plt.plot(freqs2, 20*np.log10(abs(X2_rect)+1e-6), label='Rectangular Window')
plt.plot(freqs2, 20*np.log10(abs(X2_ham)+1e-6), label='Hamming Window')
plt.title("Spectral Leakage Comparison (Composite Signal)")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Magnitude (dB)")
plt.xlim(-100, 50)
plt.legend()
plt.grid(True)
plt.show()
```