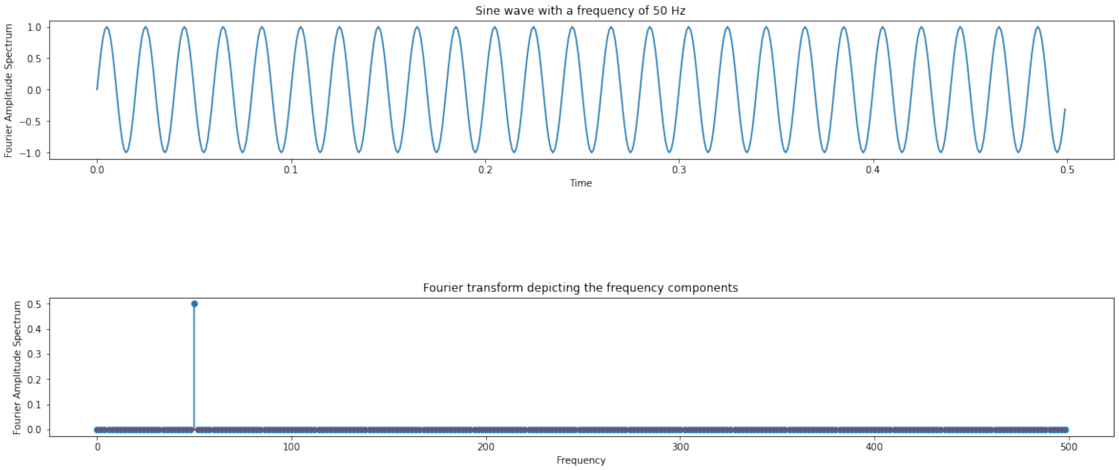
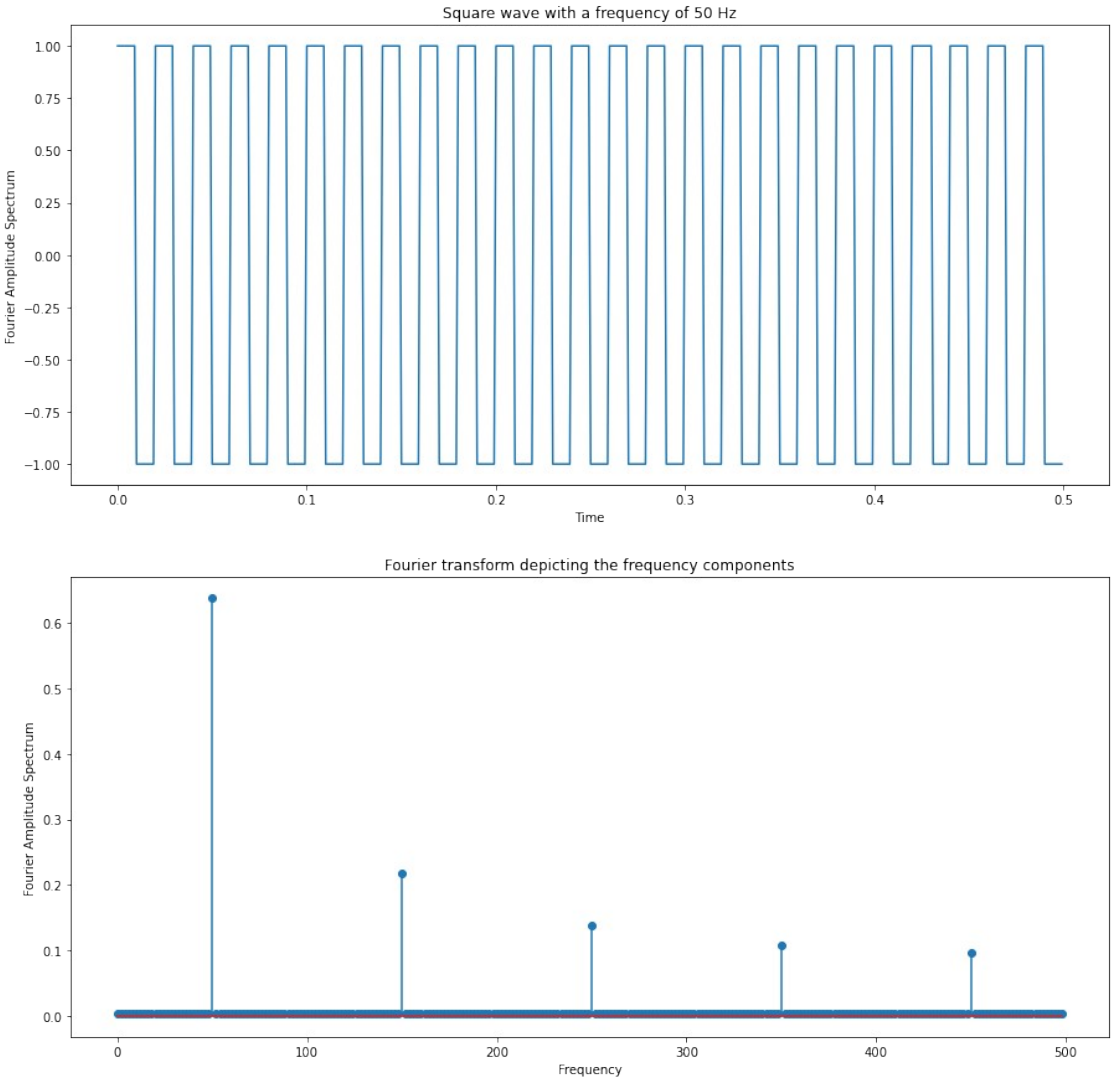
```python
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

#Question 1
#Plot the Sine Wave and Fourier amplitude Spectrum of frequency 50Hz,
sampled at 1ms.


samplingFrequency = 1000;
samplingInterval = 1 / samplingFrequency;
beginTime = 0;
endTime = 0.5;
signal1Frequency = 50;
time = np.arange(beginTime, endTime, samplingInterval);

# Using the Sin function from the numpy library to plot the sin wave
amplitude1 = np.sin(2*np.pi*signal1Frequency*time)

figure, axis = plt.subplots(2, 1,figsize=(20,8))

plt.subplots_adjust(hspace=1)

# Time domain representation for sine wave

axis[0].set_title('Sine wave with frequency of %d Hz'
%signal1Frequency)

axis[0].plot(time, amplitude1)
axis[0].set_xlabel('Time')

axis[0].set_ylabel('Fourier Amplitude Spectrum')

# Frequency domain representation

# Here we have used the fast fourier transform equation using the fft
function

fourierTransform = np.fft.fft(amplitude1)/len(amplitude1)
fourierTransform = fourierTransform[range(int(len(amplitude1)/2))]
tpCount = len(amplitude1)
values = np.arange(int(tpCount/2))
timePeriod = tpCount/samplingFrequency
frequencies = values/timePeriod

# Frequency domain representation

axis[1].set_title('Fourier transform depicting frequency components')
axis[1].stem(frequencies, abs(fourierTransform))
```

```
axis[1].set_xlabel('Frequency')
axis[1].set_ylabel('Fourier Amplitude Spectrum')
plt.show()
```



```
#Question 2
#Plot the Square Wave and Fourier amplitude Spectrum of frequency
50Hz, sampled at 1ms.


samplingFrequency = 1000;
samplingInterval = 1 / samplingFrequency;
beginTime = 0;
endTime = 0.5;
signal1Frequency = 50;
time = np.arange(beginTime, endTime, samplingInterval);

# Using the square function from the signal library to plot the square
wave

amplitude = signal.square(2*np.pi*signal1Frequency*time)
figure, axis = plt.subplots(2, 1,figsize=(15,15))

# Time domain representation for sine wave
axis[0].set_title('Square wave with a frequency of %d Hz'
%signal1Frequency)
axis[0].plot(time, amplitude)
axis[0].set_xlabel('Time')
axis[0].set_ylabel('Fourier Amplitude Spectrum')

# Frequency domain representation

# Here we have used the fast fourier transform function using the fft
equation
```

```
fourierTransform = np.fft.fft(amplitude)/len(amplitude)
fourierTransform = fourierTransform[range(int(len(amplitude)/2))]
tpCount = len(amplitude)
values = np.arange(int(tpCount/2))
timePeriod = tpCount/samplingFrequency
frequencies = values/timePeriod

# Frequency domain representation
axis[1].set_title('Fourier transform depicting the frequency
components')
axis[1].stem(frequencies, abs(fourierTransform))
axis[1].set_xlabel('Frequency')
axis[1].set_ylabel('Fourier Amplitude Spectrum')
plt.show()
```


Square wave with a frequency of 50 Hz
Fourier Amplitude Spectrum
Time
Fourier transform depicting the frequency components
Fourier Amplitude Spectrum
Frequency