In [21]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.io.wavfile import read, write
Fs,data_ = read('test_audio.wav')
```

In [2]:
```
from IPython.display import Audio
Audio('test_audio.wav')
```

Out[2]:

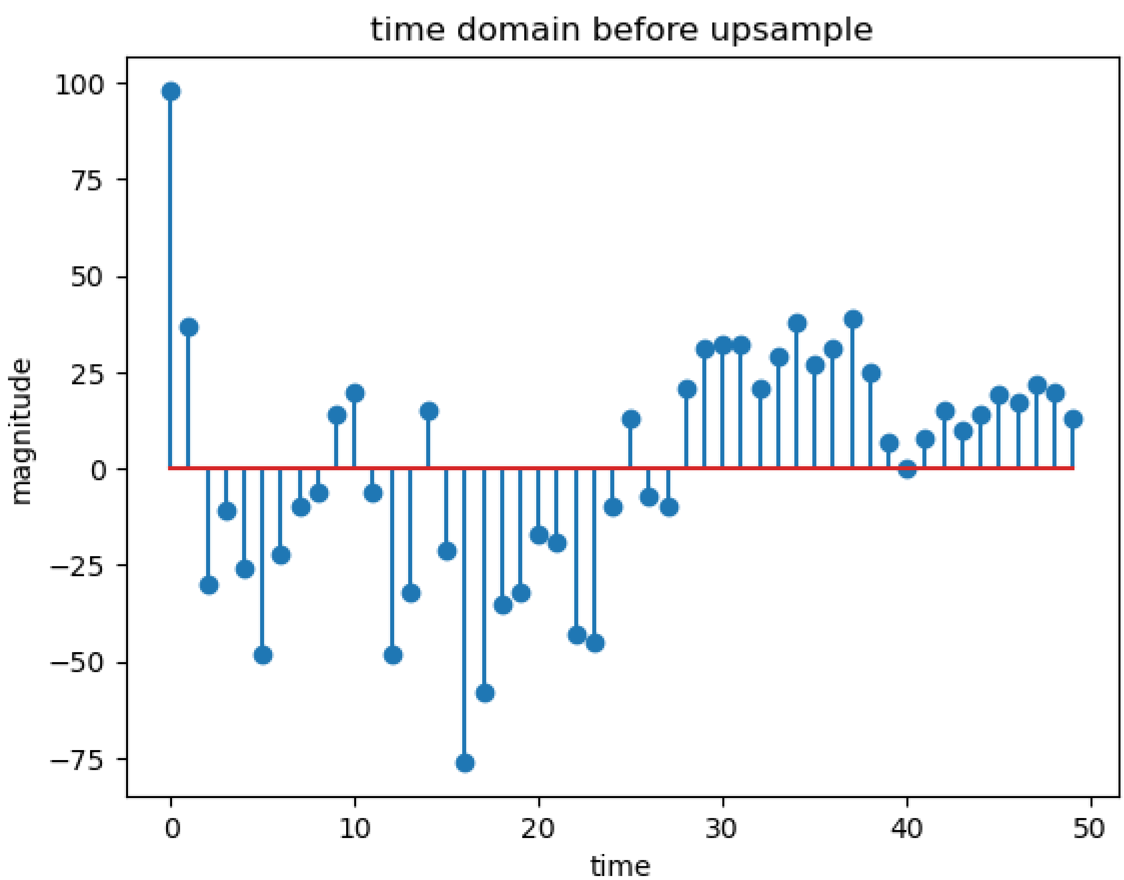
0:20 / 0:20

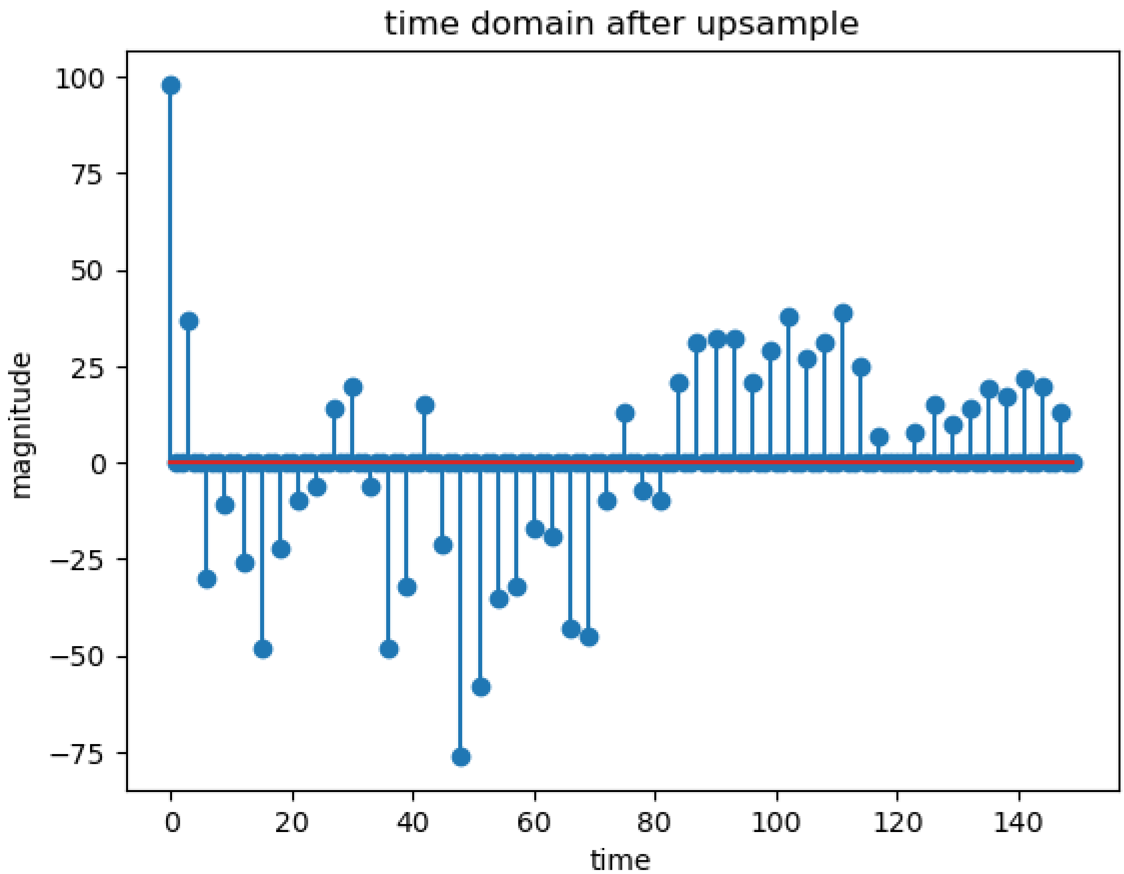
In [22]:
```
print ("Fs =",Fs)
data = data_[:,0]
data_new =[]
for i in range(len(data)):
    data_new.append(data[i])
    data_new.append(0)
    data_new.append(0)


data_fd = np.fft.fft(data)
data_new_fd = np.fft.fft(data_new)
freq_i = np.fft.fftfreq(len(data),d=1/Fs)
freq_n = np.fft.fftfreq(len(data_new),d=1/Fs)
```

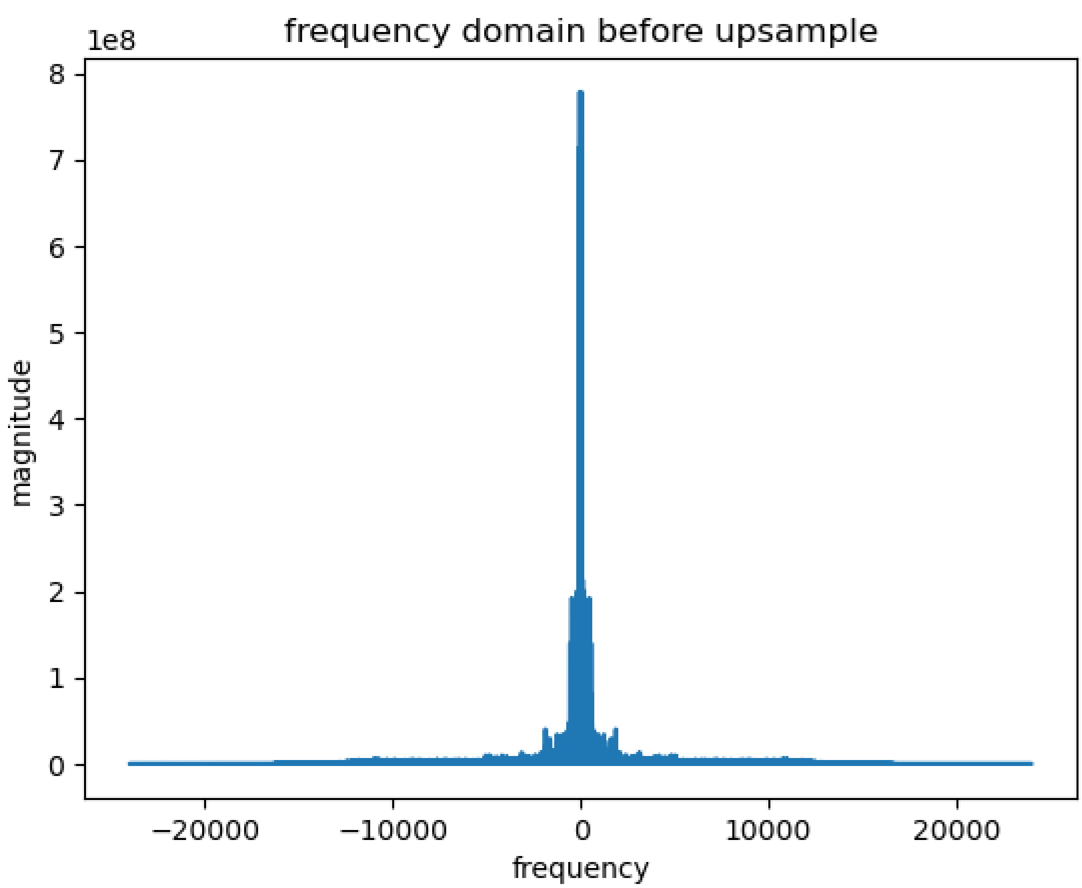
```
Fs = 48000
```

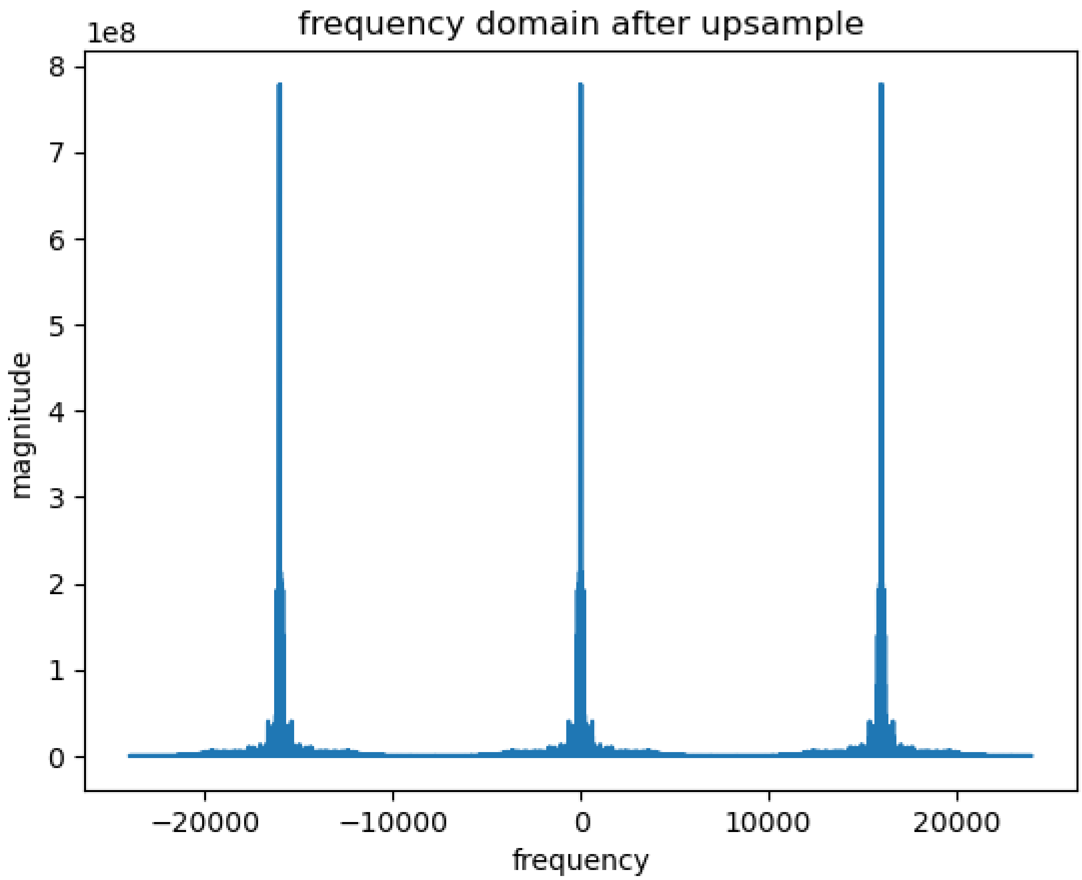
In [27]:
```
plt.figure()
plt.stem(data[-50:])
plt.xlabel('time')
plt.ylabel('magnitude')
plt.title('time domain before upsample')

plt.figure()
plt.stem(data_new[-150:])
plt.xlabel('time')
plt.ylabel('magnitude')
plt.title('time domain after upsample')

plt.figure()
plt.plot(freq_i,np.abs(data_fd))
plt.xlabel('frequency')
plt.ylabel('magnitude')
plt.title('frequency domain before upsample')

plt.figure()
plt.plot(freq_n,np.abs(data_new_fd))
plt.xlabel('frequency')
plt.ylabel('magnitude')
plt.title('frequency domain after upsample')
```

Out[27]:
```
Text(0.5, 1.0, 'frequency domain after upsample')
```




time domain after upsample
magnitude
time
100
75
50
25
0
−25
−50
−75
0
20
40
60
80
100
120
140


frequency domain before upsample
1e8
magnitude
frequency
8
7
6
5
4
3
2
1
0
−20000
−10000
0
10000
20000

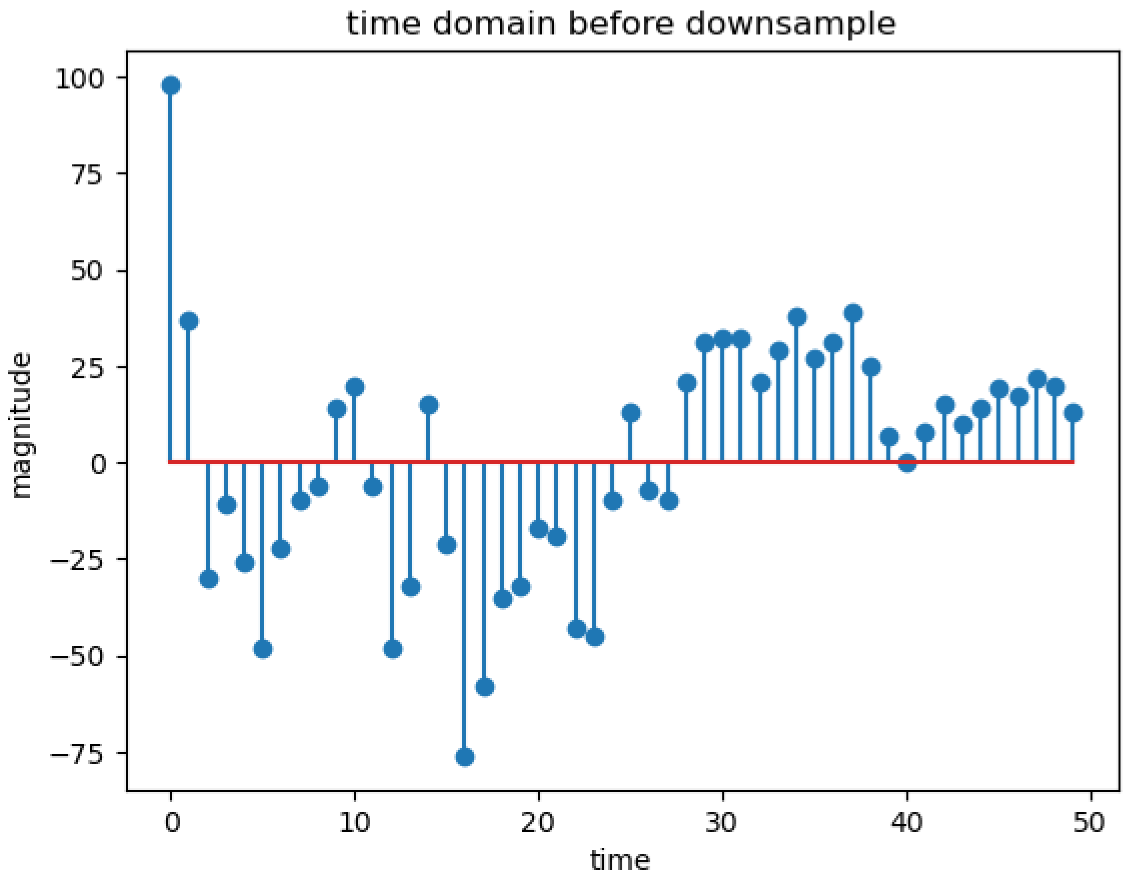


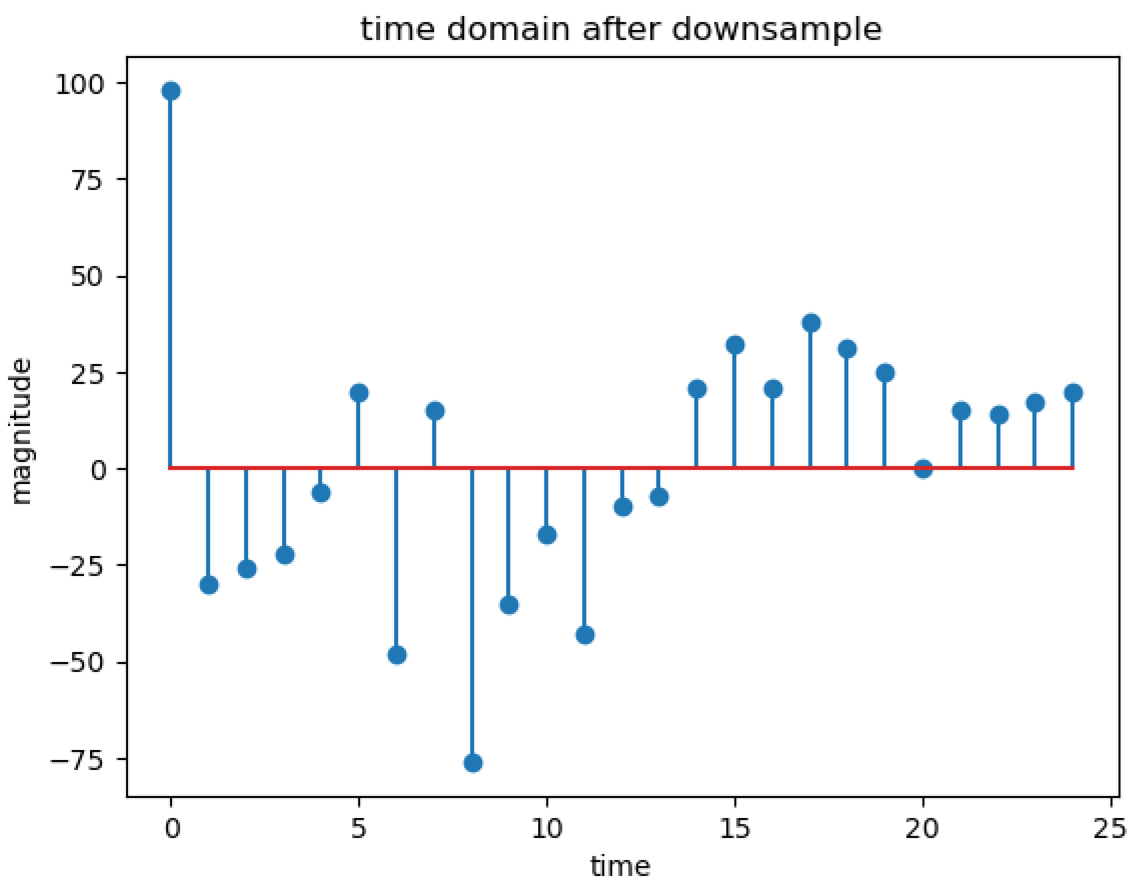
Question:

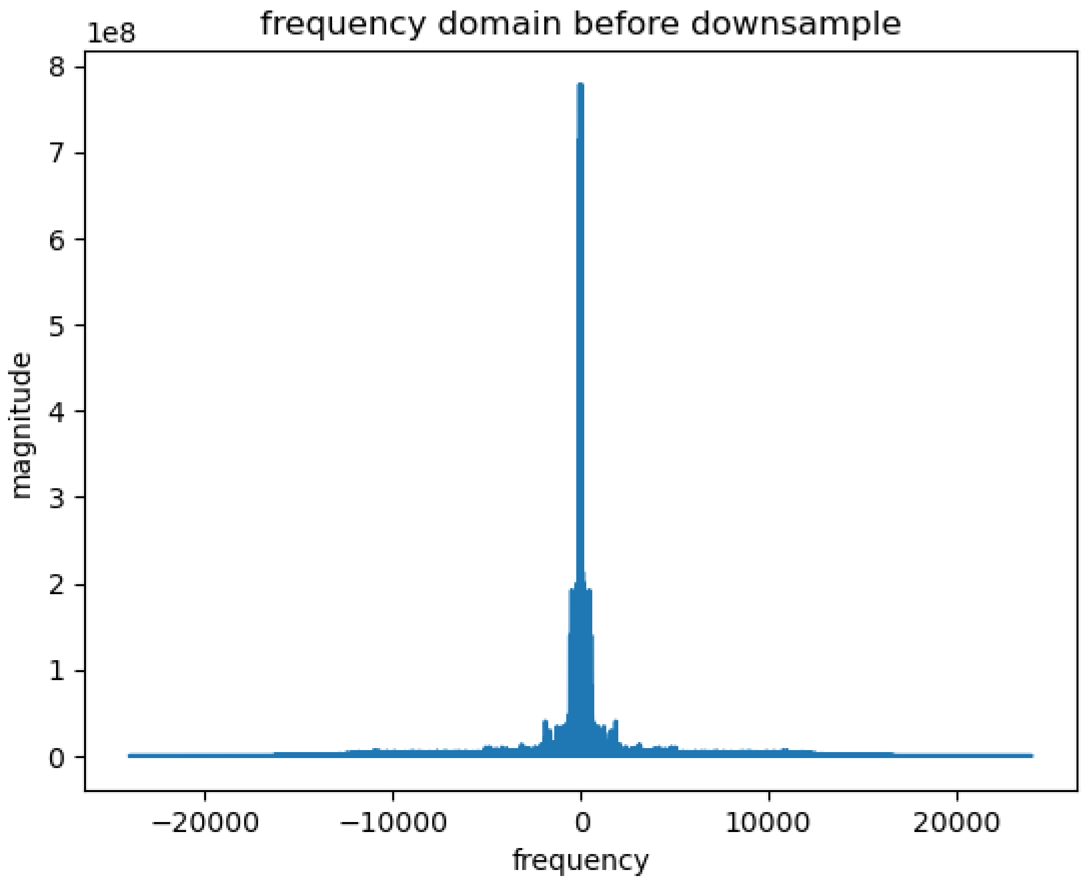
```
1. the FFT is compressed by a factor of 3, as DFT is periodic, there
are frequency leakage (aliasing)

2. we can apply a low pass filter to the upsampled data with cut off
frequency at sample_frq/3
```

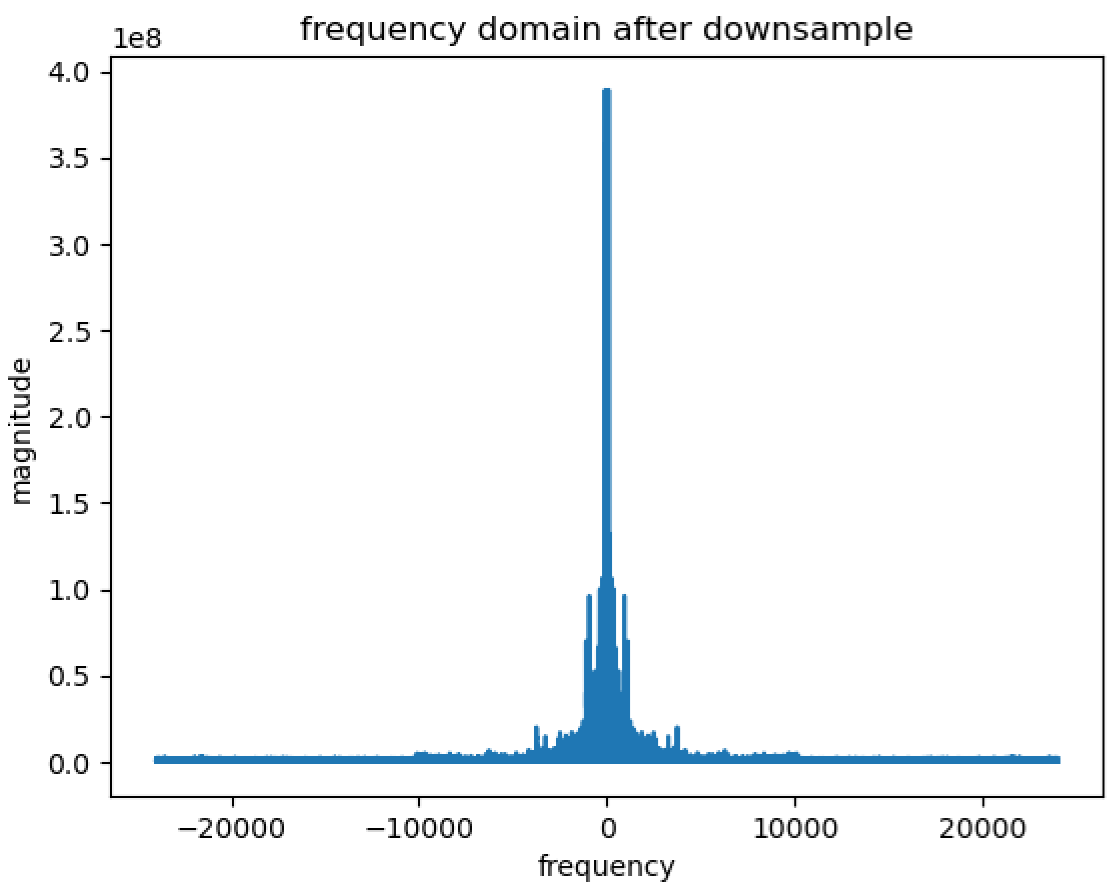
In [30]:
```
data_dn=[]
for i in range (int(len(data)/2)):
    data_dn.append(data[2*i])

data_dn_f = np.fft.fft(data_dn)
freq_dn = np.fft.fftfreq(len(data_dn_f),d=1/Fs)

plt.figure()
plt.stem(data[-50:])
plt.xlabel('time')
plt.ylabel('magnitude')
plt.title('time domain before downsample')

plt.figure()
plt.stem(data_dn[-25:])
plt.xlabel('time')
plt.ylabel('magnitude')
plt.title('time domain after downsample')

plt.figure()
plt.plot(freq_i,np.abs(data_fd))
plt.xlabel('frequency')
plt.ylabel('magnitude')
plt.title('frequency domain before downsample')

plt.figure()
plt.plot(freq_dn,np.abs(data_dn_f))
plt.xlabel('frequency')
plt.ylabel('magnitude')
plt.title('frequency domain after downsample')
```

Out[30]:
```
Text(0.5, 1.0, 'frequency domain after downsample')
```


time domain before downsample
magnitude
time
100
75
50
25
0
−25
−50
−75
0
10
20
30
40
50


time domain after downsample
magnitude
time
100
75
50
25
0
−25
−50
−75
0
5
10
15
20
25


frequency domain before downsample
1e8
8
7
6
5
4
3
2
1
0
magnitude
−20000
−10000
0
10000
20000
frequency

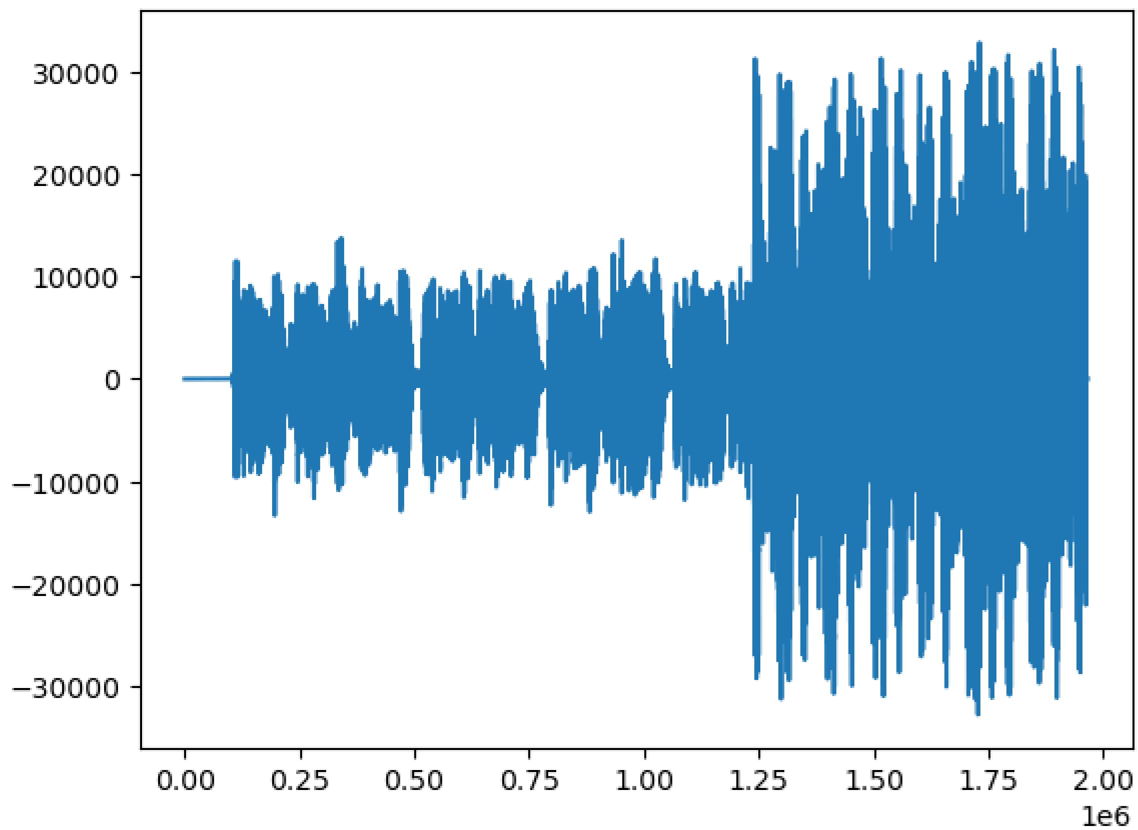

frequency domain after downsample
1e8
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
magnitude
−20000
−10000
0
10000
20000
frequency

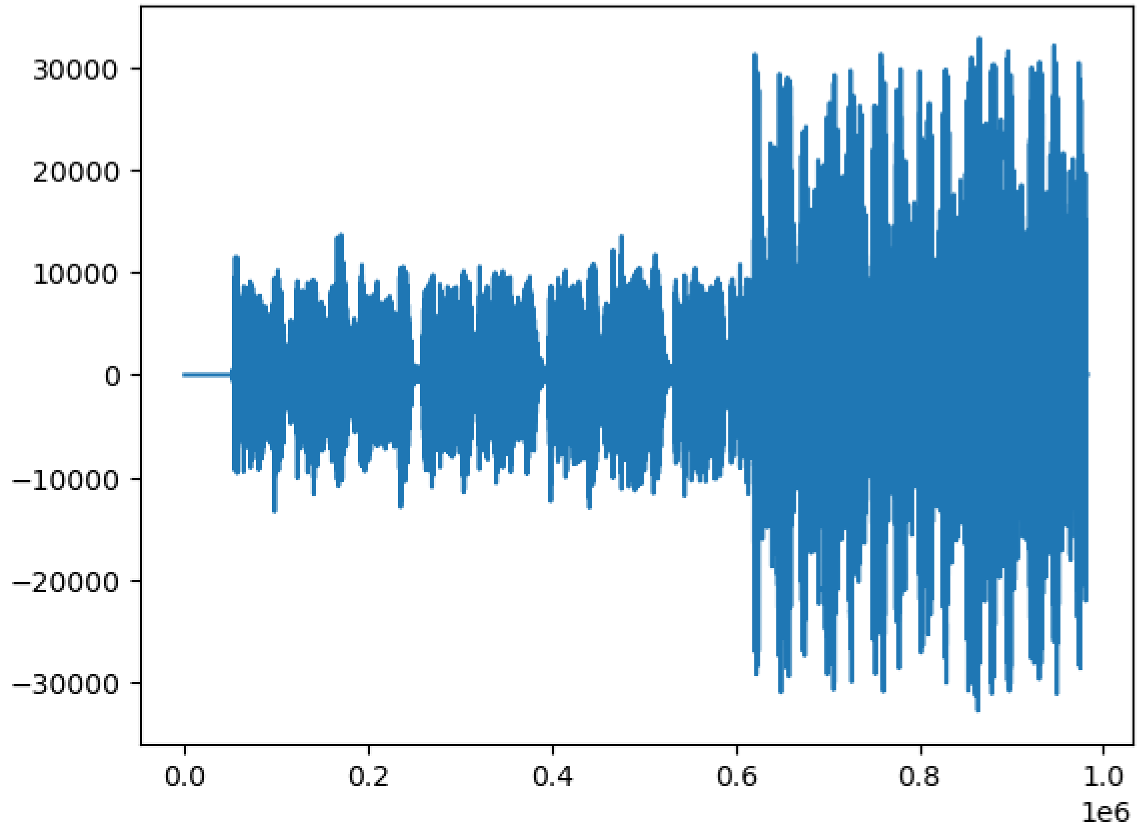
Question:

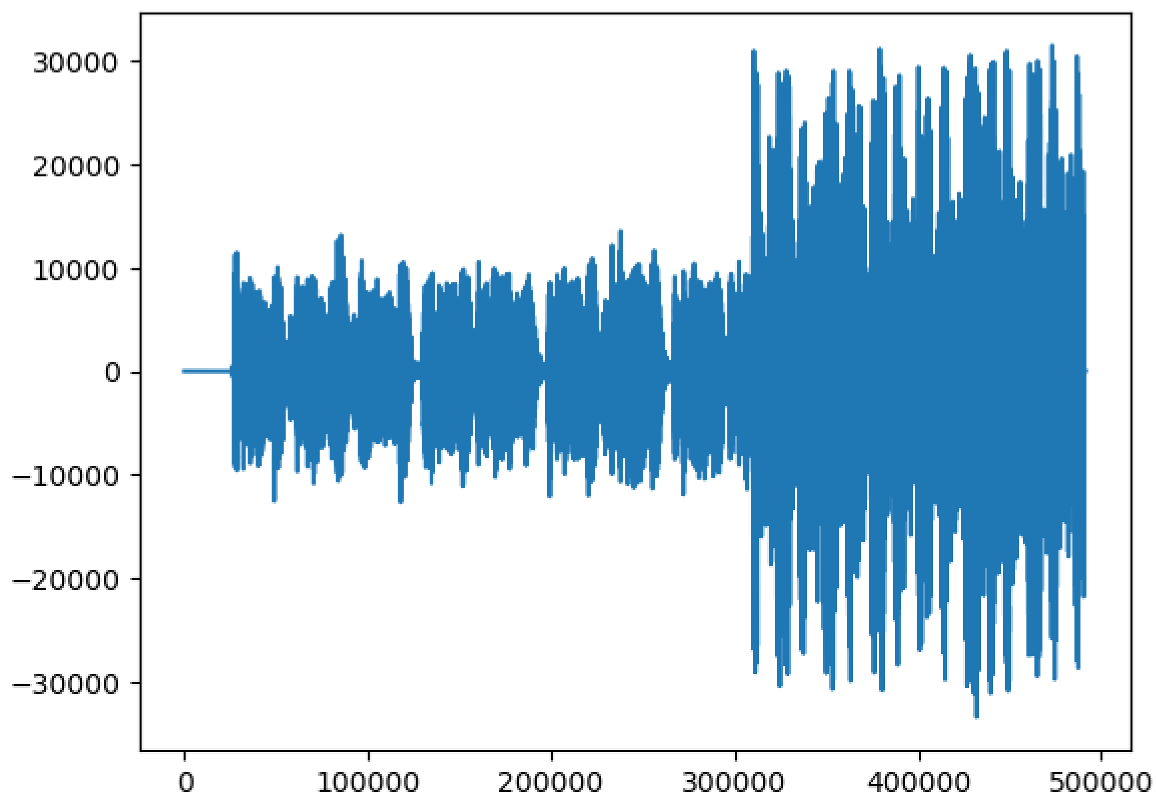
1. the downsampled signal's fft was streched by a factor of 2

2. we can apply an anti-aliasing filter before downsampling (LPF)

In [41]:
```
from scipy import signal
from scipy.io.wavfile import read
from IPython.display import Audio

Fs, data = read('test_audio.wav')
data = data[:, 0]
data = data.astype('float')

def get_output(up_,down_):


    up_ratio = up_
    down_ratio = down_

    output = signal.resample_poly(data, up_ratio, down_ratio)
    plt.figure()
    plt.plot(output)
    return output


output1 = get_output(4,2)
output2 = get_output(1,1)
output3 = get_output(2,4)
```




30000
20000
10000
0
−10000
−20000
−30000
0.0
0.2
0.4
0.6
0.8
1.0
1e6


30000
20000
10000
0
−10000
−20000
−30000
0
100000
200000
300000
400000
500000

In [42]:
```
Audio(output1, rate=Fs)
```

Out[42]:

0:01 / 0:40

In [43]:
```
Audio(output2, rate=Fs)
```

Out[43]:

0:02 / 0:20

In [44]:
```
Audio(output3, rate=Fs)
```

Out[44]:

0:10 / 0:10

Question:

```
If we have a net upsampling effect (M/N >1) we will have a "slow play
back" effect on the audio where the pitch gets lower and sound more l
ike a male.

If we have a net downsampling effect (M/N <1) we will have a "fast pl
ayback" effect on the audio where the pitch gets higher and some arti
facts are intruduced.
```