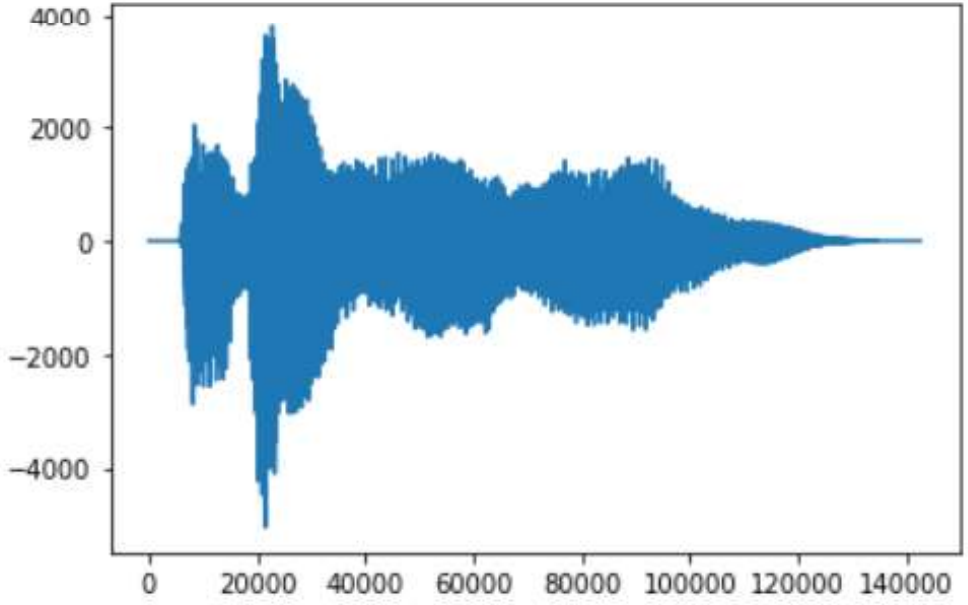
# Example

```
import array
import matplotlib.pyplot as plt

with open('C:/temp/python_wav/tada.wav','rb') as fp:
    fp.seek(0x28)
    num_bytes = int.from_bytes(fp.read(4),'little')
    print(num_bytes)
    samples = array.array('h')
    samples.fromfile(fp, num_bytes // 2 )
    plt.plot(samples)
    plt.show()
    print('end')
```

```
285184
```




```
end
```

Napisz program do wyświetlenia wykresu pliku wav 16 bitowego za pomocą modułu matplotlib.

# Budowa pliku dźwiękowego - wav

| Positions | Sample Value | Description |
|---|---|---|
| 1 - 4 | "RIFF" | Marks the file as a riff file. Characters are each 1 byte long. |
| 5 - 8 | File size (integer) | Size of the overall file - 8 bytes, in bytes (32-bit integer). Typically, you'd fill this in after creation. |
| 9 -12 | "WAVE" | File Type Header. For our purposes, it always equals "WAVE". |
| 13-16 | "fmt " | Format chunk marker. Includes trailing null |
| 17-20 | 16 | Length of format data as listed above |
| 21-22 | 1 | Type of format (1 is PCM) - 2 byte integer |
| 23-24 | 2 | Number of Channels - 2 byte integer |
| 25-28 | 44100 | Sample Rate - 32 byte integer. Common values are 44100 (CD), 48000 (DAT). Sample Rate = Number of Samples per second, or Hertz. |
| 29-32 | 176400 | (Sample Rate * BitsPerSample * Channels) / 8. |
| 33-34 | 4 | (BitsPerSample * Channels) / 8.1 - 8 bit mono2 - 8 bit stereo/16 bit mono4 - 16 bit stereo |
| 35-36 | 16 | Bits per sample |
| 37-40 | "data" | "data" chunk header. Marks the beginning of the data section. |
| 41-44 | File size (data) | Size of the data section. |
| Sample values are given above for a 16-bit stereo source. | | |

```python
import array
import matplotlib.pyplot as plt

with open('C:/temp/python_wav/tada.wav','rb') as fp:
    fp.seek(0x28)
    num_bytes = int.from_bytes(fp.read(4),'little')
    print(num_bytes)
    samples = array.array('h')
    samples.fromfile(fp, num_bytes // 2 )
    plt.plot(samples)
    plt.show()
    print('end')
```

| Type code | C Type | Python Type | Minimum size in bytes |
|---|---|---|---|
| 'b' | signed char | int | 1 |
| 'B' | unsigned char | int | 1 |
| 'u' | Py_UNICODE | Unicode character | 2 |
| 'h' | signed short | int | 2 |
| 'H' | unsigned short | int | 2 |
| 'i' | signed int | int | 2 |
| 'I' | unsigned int | int | 2 |
| 'l' | signed long | int | 4 |
| 'L' | unsigned long | int | 4 |
| | signed long long | int | 8 |
| | unsigned long long | int | 8 |
| | float | float | 4 |
| | double | float | 8 |

```
>>> int.from_bytes(b'\x00\x10', byteorder='big')
16
>>> int.from_bytes(b'\x00\x10', byteorder='little')
4096
>>> int.from_bytes(b'\xfc\x00', byteorder='big', signed=True)
-1024
>>> int.from_bytes(b'\xfc\x00', byteorder='big', signed=False)
64512
>>> int.from_bytes([255, 0, 0], byteorder='big')
16711680
```