

### Zadanie 1.

Funkcja S2BS:

```
def S2BS(s):
    result = []
    for c in s:
        bit = bin(ord(c))[2:]
        if len(bit) < 8:
            bit = '0' + bit
        result.extend([int(x) for x in bit])
    return result
```

Przykład strumienia binarnego:

```
S2BS('text')
# [0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1,
1, 1, 0, 1, 0, 0]
```

### Zadanie 2.

Funkcja obliczająca  $Z_a(t)$

$$z_A(t) = \begin{cases} A_1 \cdot \sin(2\pi ft + \varphi) & \text{dla } \overline{m}(t) = 0 \\ \underbrace{A_2 \cdot \sin(2\pi ft + \varphi)}_{s_{A_2}(t)} & \text{dla } \overline{m}(t) = 1 \end{cases}$$

```
def za1(A1, f, t, phi):
    return A1 * np.sin(2 * np.pi * f * t + phi)
def za2(A2, f, t, phi):
    return A2 * np.sin(2 * np.pi * f * t + phi)
```

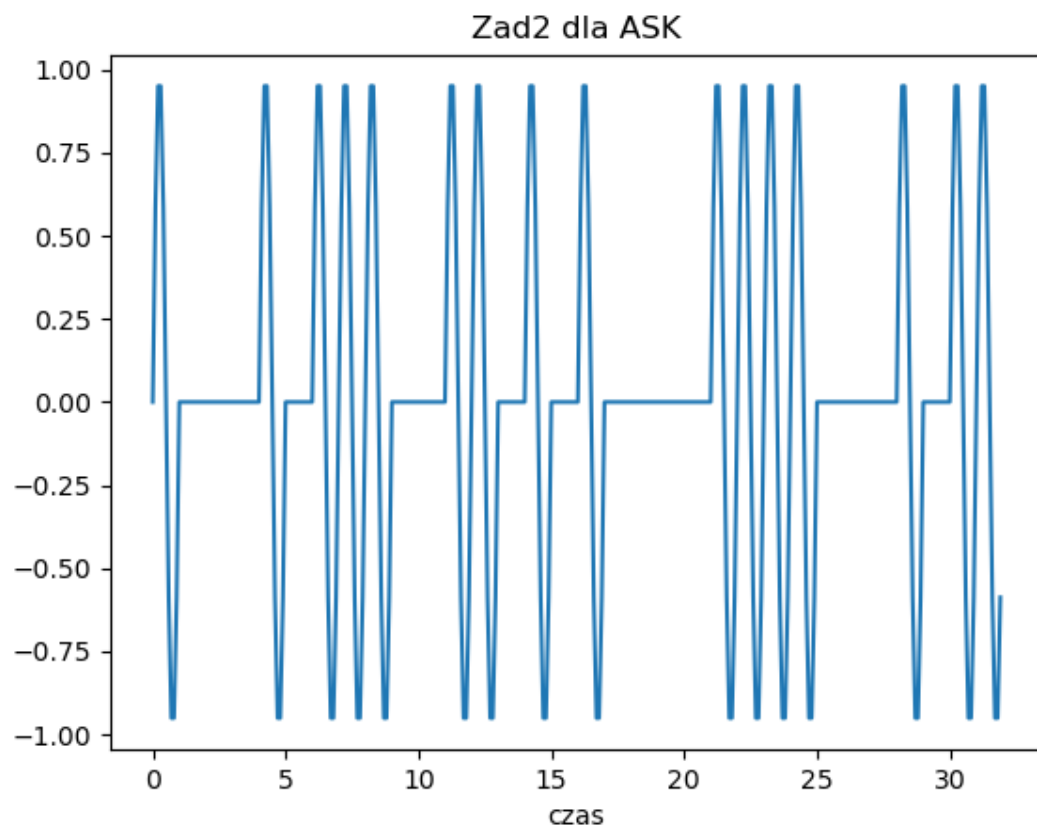
Parametry oraz funkcja do wygenerowania sygnału ASK:

```
A1 = 1
A2 = 0
Tb = 1
N = 1 / Tb
f = N / Tb
phi = 0
czas = np.arange(0, len(x), 0.1)
```

```
def sygASK():
    ASK = []
    for t in czas:
        if x[int(t)] == 0:
            ASK.append(za1(A1, f, t, phi))
        else:
            ASK.append(za2(A2, f, t, phi))
    return ASK
```

Wykres dla ASK:

```
ASK = sygASK()
plt.plot(czas, ASK)
plt.title("Zad2 dla ASK")
plt.show()
```



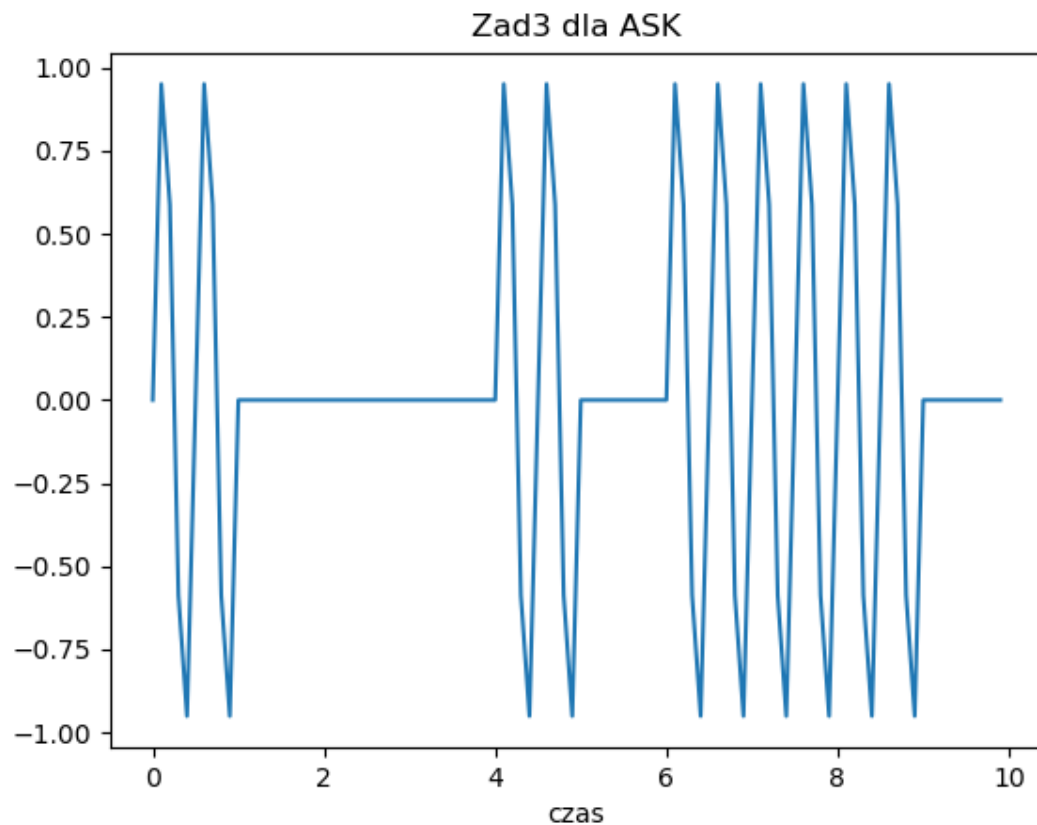
### Zadanie 3.

Sygnał zmodulowany dla  $Z_a(t)$  w dziedzinie czasu  $N=2$  oraz przy ograniczonej liczbie bitów do 10.

```
A1 = 1
A2 = 0
Tb = 1
N = 2
f = N / Tb
phi = 0
czas2 = np.arange(0, 10, 0.1)
```

```
plt.plot(czas2, wASK)
plt.title("Zad3 dla ASK")
plt.xlabel('czas')
plt.show()
```

Wykres:



#### Zadanie 4.

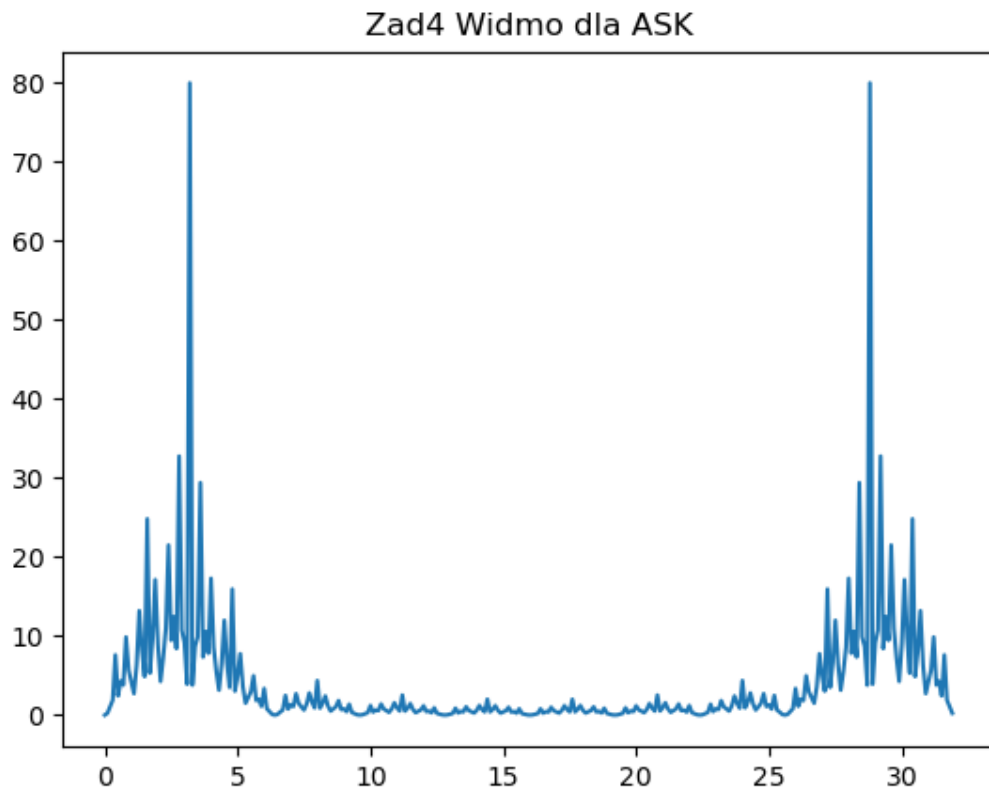
Funkcje:

```
def M(x):  
    Re = np.real(x)  
    Im = np.imag(x)  
    mk = []  
    for i in range(0, len(Re)):  
        mk.append(np.sqrt(Re[i] ** 2 + Im[i] ** 2))  
    return mk
```

```
def DFT(n):  
    xk = []  
    for i in range(len(n)):  
        temp = 0  
        for j in range(len(n)):  
            temp += n[j] * np.exp(1j * 2 * np.pi / len(n)) ** (-i * j)  
        xk.append(temp)  
    return xk
```

```
plt.plot(czas, M(DFT(ASK)))  
plt.title("Zad4 Widmo dla ASK")  
plt.show()
```

Wykres:



### Zadanie 5.

Szerokość pasma sygnału zmodulowanego dla ASK:

```
max = np.max(ASK)  
min = np.min(ASK)  
w = max - min  
# W = 1.9021130325903224
```