

Задание 1.

Создать класс CyclicCodes. Объявить и инициализировать поля

$k = 4$ (тип: целочисленное значение)

$n = 7$ (тип: целочисленное значение)

$g = [1, 0, 1, 1]$ (тип: массив, numpy-массив или bytearray).

Задание 2.

Объявить и реализовать метод класса, осуществляющий умножение входного двоичного массива длины k на образующий многочлен $g(x)$, соответствующий заданному вектору g . Результатом работы функции должен быть вектор, соответствующий произведению.

Обратите внимание, что:

Для $a(x)$:

$$a = [1000], a(x) = 1, c(x) = g(x), c = [1011000]$$

$$a = [0100], a(x) = x, c(x) = xg(x), c = [0101100]$$

$$a = [0010], a(x) = x^2, c(x) = x^2g(x), c = [0010110]$$

$$a = [0001], a(x) = x^3, c(x) = x^3g(x), c = [0001011]$$

Алгоритм умножения:

1. Преобразовать входное сообщение в многочлен по принципу:

$$a = [a_0 a_1 a_2 a_3] \rightarrow a(x) = a_0 + a_1x + a_2x^2 + a_3x^3.$$

2. Умножить на образующий:

$$c(x) = a(x)g(x).$$

3. Преобразовать обратно в сообщение:

$$c(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5 + c_6x^6 \rightarrow c = [c_0 c_1 c_2 c_3 c_4 c_5 c_6]$$

Псевдокод для умножения:

```
def Encode(self, a):  
    c = [0000000]  
    for i = 0:7  
        if (a[i])  
            c[i:i+4] = c[i:i+4] + g  
    return c
```

Задание 3.

Объявить и реализовать метод класса, осуществляющий деление с остатком входного двоичного массива длины n на образующий многочлен $g(x)$, соответствующий заданному вектору g .

Результатом работы функции должен быть вектор, соответствующий остатку от деления.

Алгоритм деления с остатком:

1. Преобразовать входное сообщение в многочлен по принципу:

$$c = [c_0 c_1 c_2 c_3 c_4 c_5 c_6] \rightarrow c(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5 + c_6x^6.$$

2. Вычислить остаток от деления на образующий:

$$r(x) = c(x) \bmod g(x).$$

3. Преобразовать обратно в сообщение:

$$r(x) = r_0 + r_1x + r_2x^2 \rightarrow r = [r_0 r_1 r_2]$$

Псевдокод для деления с остатком:

```
def Remainder(self, c):  
    rem = c  
    for i = n:n-k
```

```

        if (rem[i])
            rem[i-(n - k):i+1] = rem[i-(n - k):i+1] + g
    return rem

```

Задание 4.

Объявить и реализовать метод класса, осуществляющий систематическое кодирование входного двоичного массива длины k . Результатом работы функции должен быть вектор, соответствующий полученному кодовому сообщению

Алгоритм систематического кодирования:

1. Преобразовать входное сообщение в многочлен по принципу:

$$a = [a_0 a_1 a_2 a_3] \rightarrow a(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3.$$

2. Сдвинуть вправо на $n-k$ разрядов :

$$c'(x) = x^3 a(x).$$

3. Вычислить остаток от деления на образующий:

$$r(x) = c'(x) \bmod g(x)$$

3. Сложить сообщение со сдвигом с остатком:

$$c(x) = c'(x) + r(x)$$

3. Преобразовать обратно в сообщение:

$$c(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 + c_5 x^5 + c_6 x^6 \rightarrow c = [c_0 c_1 c_2 c_3 c_4 c_5 c_6]$$

```

def EncodeSys(self, a):
    c = [00000000]
    c[n-k : n] = a
    r = Remainder(c)
    c[0 : n-k] = r
    return c

```

Задание 5 (необязательное).

Объявить и реализовать методы класса, осуществляющие кодирование и декодирование файлов на диске с обнаружением ошибки посредством циклических кодов.