**Лабораторная работа 4.**

**Часть 4.1.**

**Задание 4.1.1.**

Создать класс GolayCode. Объявить и инициализировать поля

**число информационных (исходных) разрядов** k = 12 (константа, тип: целочисленное значение)

**длина закодированного сообщения** n = 24 (константа, тип: целочисленное значение)

**матрица B (см. лекции)** (тип: массив, numpy-массив или bitarray).

**Задание 4.1.2.**

Объявить и реализовать метод класса, осуществляющий кодирование входной битовой последовательности посредством умножения входного двоичного массива длины k на порождающую матрицу .

**Задание 4.1.3.**

Объявить и реализовать метод класса, осуществляющий обнаружение и исправление ошибок (с использованием алгоритма из лекций). Проверить работоспособность и сравнить результаты с примерами из лекций.

**Часть 4.2.**

**Задание 4.2.1.**

Создать класс RMCode. Объявить и инициализировать поля

**r** и **m** (тип: целочисленные значения, ).

**число информационных (исходных) разрядов** k =  (тип: целочисленное значение)

**длина закодированного сообщения** n = 2m (тип: целочисленное значение)

**Задание 4.2.2.**

Объявить и реализовать метод класса, вычисляющий порождающую матрицу **G** (в соотвествии с рекурсивным алгоритмом из лекций), и метод, осуществляющий кодирование входной битовой последовательности посредством умножения входного двоичного массива длины k на порождающую матрицу. Сравнить полученные матрицы с примерами из лекций. Проверить работоспособность кодировщика для **r = 1, m = 3** и сравнить результаты с примерами из лекций.

**Задание 4.2.3.**

Объявить и реализовать метод класса, осуществляющий обнаружение и исправление ошибок (с использованием алгоритма из лекций). Проверить работоспособность декодира для **r = 1, m = 3** и сравнить результаты с примерами из лекций.

**Задание 4.2.4.**

Объявить и реализовать методы класса, осуществляющие кодирование и декодирование файлов на диске с обнаружением ошибки посредством кодирования и декодирования из заданий 4.2.2-4.2.3.