Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа №4

по информатике

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант 15

Выполнил:

Жеребцов М. А. Р3117

Проверил:

Марухленко Д.С.

Содержание

[Задание 3](#_Toc180177527)

[Основные этапы работы 4](#_Toc180177528)

[Дополнительное задание 8](#_Toc180177529)

[Заключение 10](#_Toc180177530)

# Обязательное задание

С помощью replace заменяю метасимволы в определенном порядке, полученным эмпирическим путем.

# Дополнительное задание

Задание: написать программу на любом языке программирования, которая на вход получает набор из 7 цифр «0» и «1», записанных подряд, анализирует это сообщение на основе классического кода Хэмминга (7,4), а затем выдает правильное сообщение (только информационные биты) и указывает бит с ошибкой при его наличии.

Решение (Рисунок 6): написал скрипт на python, который выполняет поставленную задачу.

def analyze\_message(message: str) -> tuple[str, int]:

    int\_list = list(map(int, list(message)))

    s1 = int\_list[0] ^ int\_list[2] ^ int\_list[4] ^ int\_list[6]

    s2 = int\_list[1] ^ int\_list[2] ^ int\_list[5] ^ int\_list[6]

    s3 = int\_list[3] ^ int\_list[4] ^ int\_list[5] ^ int\_list[6]

    s = str(s1) + str(s2) + str(s3)

    match s:

        case '000':

            return message[2] + message[4] + message[5] + message[6], -1

        case '001':

            return message[2] + message[4] + message[5] + message[6], 4

        case '010':

            return message[2] + message[4] + message[5] + message[6], 2

        case '011':

            return message[2] + message[4] + ('0' if message[5] == '1' else '1') + message[6], 6

        case '100':

            return message[2] + message[4] + message[5] + message[6], 1

        case '101':

            return message[2] + ('0' if message[4] == '1' else '1') + message[5] + message[6], 5

        case '110':

            return ('0' if message[2] == '1' else '1') + message[4] + message[5] + message[6], 3

        case '111':

            return message[2] + message[4] + message[5] + ('0' if message[6] == '1' else '1'), 7

if \_\_name\_\_ == '\_\_main\_\_':

    fixed\_message, error\_bit = analyze\_message(input('Введите сообщение: '))

    print(

        f'''

Правильное сообщение: {fixed\_message}

Номер бита с ошибкой: {"ошибок не обнаружено" if error\_bit == -1 else error\_bit}

        '''

    )

Рисунок 6. Листинг кода. Анализатор кода Хэмминга

# Заключение

Я научился декодировать классический код Хемминга. Изучил схему декодирования (7;4) и (15;11) кода Хемминга. Научился вычислять минимальное число проверочных разрядов и коэффициент избыточности. Применил новые знания при написании скрипта для дополнительного задания.