

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ «Санкт-Петербургский национальный исследовательский  
университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ  
ТЕХНИКИ

## **ЛАБОРАТОРНАЯ РАБОТА №2**

по дисциплине

«ИНФОРМАТИКА»

Вариант № 70

***Выполнил:***

Студент группы Р3116

Билошицкий Михаил Владимирович

***Преподаватель:***

Машина Екатерина Алексеевна

Санкт-Петербург, 2022

# Содержание

Задания.....	3
Основные этапы вычисления.....	4
Вывод.....	8

# Задания

## Основное условие

### 2.1 Порядок выполнения работы

1. Определить свой вариант задания с помощью номера в ISU (он же номер студенческого билета). Вариантом является комбинация 3-й и 5-й цифр. Т.е. если номер в ISU = 12**34**56, то вариант = 35.
2. На основании номера варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода.
3. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения.
4. Показать, исходя из выбранных вариантов сообщений (по 4 у каждого – часть №1 в варианте), имеются ли в принятом сообщении ошибки, и если имеются, то какие. **Подробно прокомментировать** и записать правильное сообщение.
5. На основании номера варианта задания выбрать 1 полученное сообщение в виде последовательности 11-символьного кода.
6. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
7. Показать, исходя из выбранного варианта сообщений (по 1 у каждого – часть №2 в варианте), имеются ли в принятом сообщении ошибки, и если имеются, то какие. **Подробно прокомментировать** и записать правильное сообщение.
8. Сложить номера всех 5 вариантов заданий. **Умножить полученное число на 4**. Принять данное число как число информационных разрядов в передаваемом сообщении. Вычислить для данного числа минимальное число проверочных разрядов и коэффициент избыточности.
9. Необязательное задания для получения оценки «5» (позволяет набрать от 86 до 100 процентов от максимального числа баллов БаРС за данную лабораторную). Написать программу на любом языке программирования, которая на вход из командной строки 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.

### Задание 1-2

70	52	89	14	11	20
----	----	----	----	----	----

### Задание 1 (коды)

1	52	1	0	1	1	0	1	1
1	89	0	1	0	1	1	1	0
1.	14	1	1	1	1	0	0	0
1.	11	1	0	1	1	0	0	0

### Задание 2 (код)

2.	20	0	1	1	0	0	0	1	0	1	0	0	0	0	0	1
----	----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

# ОСНОВНЫЕ ЭТАПЫ ВЫЧИСЛЕНИЯ

Для вычислений использовалась программа, написанная на языке Python, в которой был реализован алгоритм кодирования в код Хэмминга и Декодирования из него. Программа автоматически выводит информацию с исходным сообщением с исправленной ошибкой и номер ошибочного бита.

Код программы:

```
def insert_symbol(my_str, sybmol, index):
    return my_str[:index] + sybmol + my_str[index:]

def replace_by_index(my_str, symbol, index):
    return my_str[:index] + symbol + my_str[index + 1:]

# encode_shred Закодировать сообщение в код Хэмминга
# Принимает на вход строку в бинарном виде
# Возвращает двоичный код с подсчитанными контрольными битами
def encode_shred(binary_input):
    power = 0
    control_byte_index = 2 ** power - 1
    encoded = binary_input
    while control_byte_index < len(encoded):
        encoded = insert_symbol(encoded, '-', control_byte_index)
        power += 1
        control_byte_index = 2 ** power - 1
    power = -1
    for k in range(len(encoded)):
        if encoded[k] == '-':
            power += 1
            byte_value = 0
            cur_index = k
            while cur_index <= len(encoded):
                start = cur_index
                end = cur_index + (2 ** power)
                for i in range(start, end):
                    cur_index += 1
                    if i >= len(encoded):
                        break
                    if encoded[i] == '-':
                        continue
                    else:
                        byte_value += int(encoded[i])
                cur_index += (2 ** power)
            encoded = encoded.replace('-', str(byte_value % 2), 1)
    return encoded

# decode_shred Отобразить информационные биты
# Принимает на вход двоичный код Хэмминга
```

```

# Возвращает 2 переменных,
# 1. Исходное сообщение с исправленной ошибкой,
# 2. Сообщение с кодом Хэмминга,
# 3. Бит с ошибкой (если нет, то = 0)
def decode_shred(binary_input):
    power = 0
    control_bytes = []
    while (2 ** power - 1) < len(binary_input):
        byte_value = 0
        cur_index = (2 ** power - 1)
        while cur_index <= len(binary_input):
            start = cur_index
            end = cur_index + (2 ** power)
            for i in range(start, end):
                cur_index += 1
                if i >= len(binary_input):
                    break
                if i + 1 == 2 ** power:
                    continue
                byte_value += int(binary_input[i])
            cur_index += (2 ** power)
        control_bytes.append(byte_value % 2)
        power += 1

    power = 0
    error_byte_n = 0
    i = 0
    while (2 ** power - 1) < len(binary_input):
        if int(binary_input[2 ** power - 1]) != control_bytes[i]:
            error_byte_n += 2 ** power
            i += 1
        power += 1
    ch_binary_input = binary_input
    if error_byte_n != 0:
        ch_binary_input = replace_by_index(binary_input, str(abs(
            int(binary_input[error_byte_n - 1]) - 1)),
        error_byte_n - 1)
    message = ''
    power = 0
    for i in range(len(ch_binary_input)):
        if i + 1 != 2 ** power:
            message += ch_binary_input[i]
        else:
            power += 1
    return message, binary_input, error_byte_n

d = ''
while not (d in ['1', '2', '3', '4']):
    d = input('Введите число от 1 до 2
1. Закодировать сообщение в код Хэмминга
2. Раскодировать сообщение из кода Хэмминга
-> ')
    if d == '1':

```

```

try:
    i = input('Введите двоичный код -> ')
    int(i, 2)
    print("Сообщение в коде Хэмминга:", encode_shred(i))
except:
    print('Неверный формат ввода данных')
elif d == '2':
    try:
        i = input('Введите двоичный код -> ')
        int(i, 2)
        o1, o2, o3 = decode_shred(i)
        print()
        print('Исходное сообщение с исправленной ошибкой:', o1)
        print('Сообщение с кодом Хэмминга:', o2)
        print('Бит с ошибкой (если нет, то = 0):', o3)
    except:
        print('Неверный формат ввода данных')

```

Решение заданий 1-2 через программу:

Код 52, задание 1:

Введите двоичный код -> 1011011

Исходное сообщение с исправленной ошибкой: 1010

Сообщение с кодом Хэмминга: 1011011

Бит с ошибкой (если нет, то = 0): 7

Код 89, задание 1:

Введите двоичный код -> 0101110

Исходное сообщение с исправленной ошибкой: 0010

Сообщение с кодом Хэмминга: 0101110

Бит с ошибкой (если нет, то = 0): 5

Код 14, задание 1:

Введите двоичный код -> 1111000

Исходное сообщение с исправленной ошибкой: 1000

Сообщение с кодом Хэмминга: 1111000

Бит с ошибкой (если нет, то = 0): 4

Код 11, задание 1:

Введите двоичный код -> 1011000

Исходное сообщение с исправленной ошибкой: 1010

Сообщение с кодом Хэмминга: 1011000

Бит с ошибкой (если нет, то = 0): 6

Код 20, задание 2:

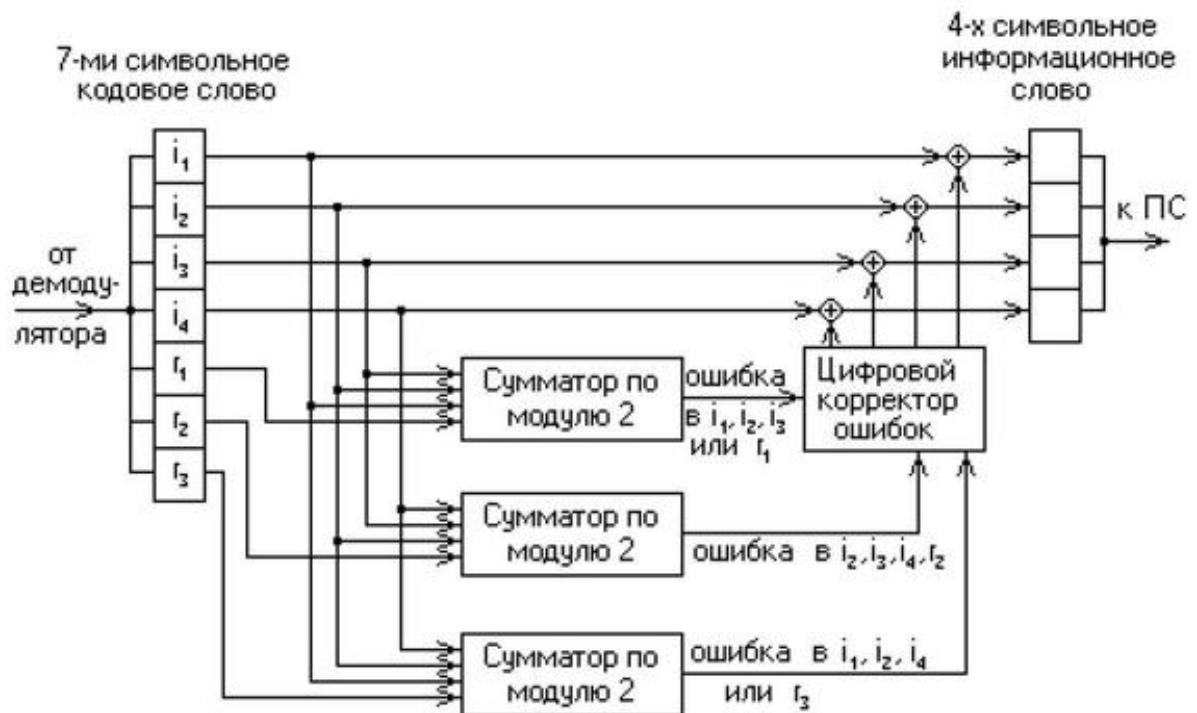
Введите двоичный код -> 011000101000001

Исходное сообщение с исправленной ошибкой: 10011000001

Сообщение с кодом Хэмминга: 011000101000001

Бит с ошибкой (если нет, то = 0): 0

Схема декодирования кода Хэмминга



Задание 8

$$(52+89+14+11+20)*4=744$$

$$2 \text{ в степени } 9 < 744 < 2 \text{ в степени } 10$$

Следовательно, будет 9 проверочных разрядов.

$$9 / (744 + 9) = 0.012 \text{ – коэффициент избыточности.}$$

## Вывод

Во время выполнения лабораторной работы я изучил работу самовосстанавливающегося кода Хэмминга, смог реализовать кодирование и декодирование программно и выполнить все задания. Полученные знания пригодятся мне в будущем.