Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Отчёт по информатике

Лабораторная работа № 2

Вариант 74

Выполнил: Бутвин.М.П, студент группы P3130

Проверил: Балакшин П.В, доцент факультета ПИиКТ

Санкт-Петербург – 2022

Оглавление

[Задание 3](#_Toc115246272)

[Основные этапы вычисления 4](#_Toc115246273)

[Вывод 5](#_Toc115246274)

[Список литературы 6](#_Toc115246275)

# Задание

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 = 123456, 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.

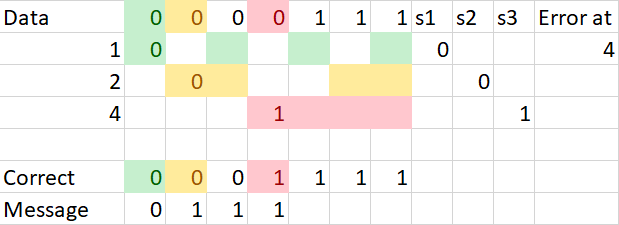
8. Сложить номера всех 5 вариантов заданий. Умножить полученное число на 4. Принять данное число как число информационных разрядов в передаваемом сообщении. Вычислить для данного числа минимальное число проверочных разрядов и коэффициент избыточности.

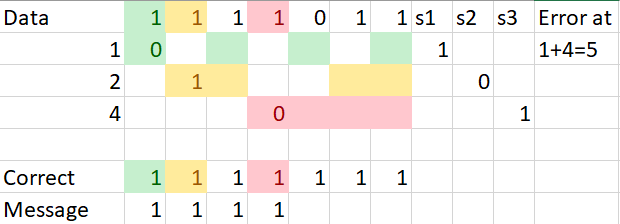
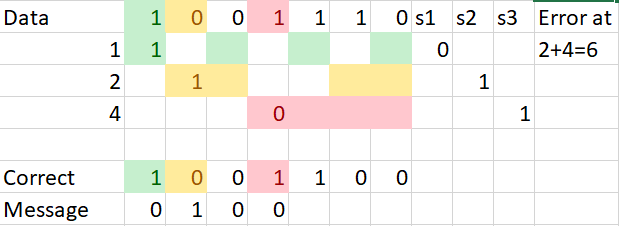
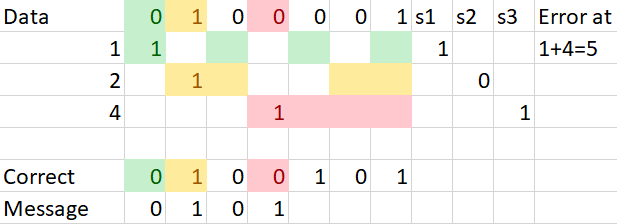
9. Необязательное задания для получения оценки «5» (позволяет набрать от 86 до 100 процентов от максимального числа баллов БаРС за данную лабораторную). Написать программу на любом языке программирования, которая на вход из командной строки получает набор из 7 цифр «0» и «1», записанных подряд, анализирует 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.

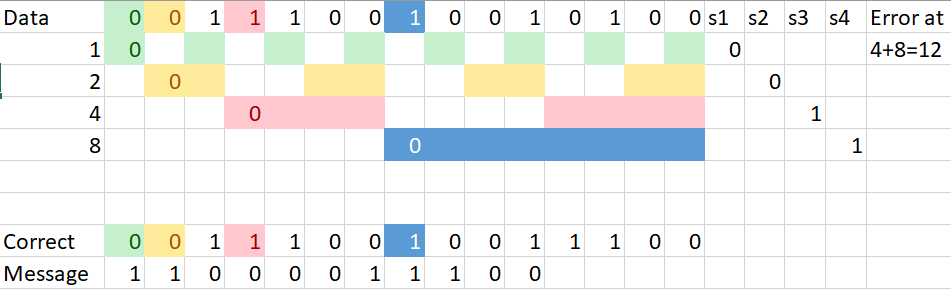
# Основные этапы вычисления

Алгоритм решения:

* Выделим цветами проверочные разряды в сообщении
* Отметим биты, за которые отвечает каждый контрольный
* Посчитаем их сумму по модулю два и запишем результат для каждого проверочного разряда
* Выполним сложение по модулю два получившихся значений, с значениями в сообщении
* Получим сигнатуру
* По сигнатуре определим позицию ошибочного бита, исправим его
* Выпишем сообщение без контрольных битов

Задание 1



Задание 2

Задание 3

(1111 + 0100 + 0101 + 0111 + 11000011100)2 \* 410 = 638010

2\*\*r >= 6380 + r + 1

min\_r = 13 – минимальное число проверочных разрядов

min\_r / (min\_r + 6380) = 0.0020334741123103394 – коэффициент избыточности

Задание 4

Код программы на Python

def decode(code: str) -> tuple[str, int]:

"""Return corrected message and position of error bit (0 if no errors)"""

if len(code) != 7:

raise ValueError('Data length mast be 7')

n\_code = [int(i) for i in code]

r1 = sum(n\_code[::2][1:]) % 2

r2 = sum(n\_code[2:3] + n\_code[5:7]) % 2

r3 = sum(n\_code[4:]) % 2

s1 = n\_code[0] != r1

s2 = n\_code[1] != r2

s3 = n\_code[3] != r3

err\_pos = s1 + s2 \* 2 + s3 \* 4

if err\_pos != 0:

n\_code[err\_pos - 1] = 1 - n\_code[err\_pos - 1]

msg = ''.join(map(str, [n\_code[2]] + n\_code[4:7]))

return msg, err\_pos

# Вывод

В ходе работы я познакомился с способами корректировки данных и поиска ошибок, методами хранения чисел в компьютере



Ричард Хэмминг, математик, создатель кода Хэмминга

# Список литературы

1. Балакшин П.В, доцент факультета ПИиКТ «Презентация по информатике» https://t.me/balakshin\_students
2. <https://ru.wikipedia.org/wiki/Код_Хэмминга>
3. <https://www.youtube.com/watch?v=cZlEeyYR4iI>