Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования   
**Национальный исследовательский университет «МЭИ»**

**Лабораторная работа №7**

по курсу«**Методы и средства защиты информации**»

**«Порождающая и проверочная матрица»**

Выполнил:

студент группы А-07-18

Востриков Р.В.

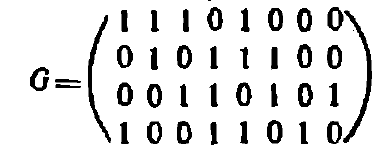
Вариант 8

Москва  
2020

**1. Задание**

Задание 1

Двоичный (8,4) код задан порождающей матрицей



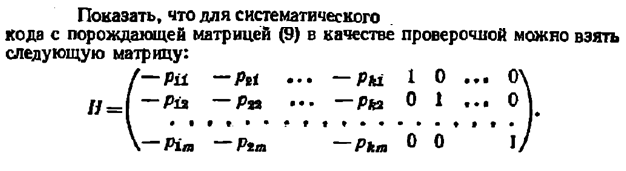
А) Составить программу, которая выводит все кодовые слова и определяет кодовое расстояние.

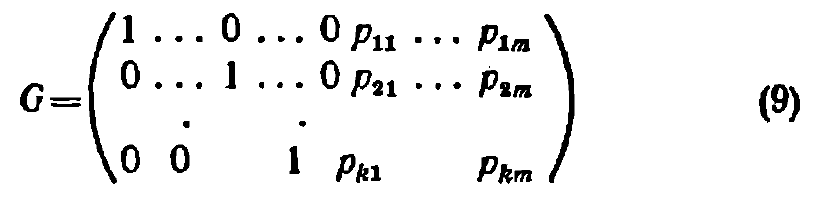
Б) Привести матрицу G с помощью эквивалентных преобразований к виду (E|-AT}

Используя соотношение А = -(-АT)T

Оттуда найти матрицу H = ( А | E ), где E- единичная матрица.

Определить произведение G и HT

Задание 2  




**2. Описание результатов**

Задание 1а)

Кодовая таблица для данной матрицы G:

0000 - 00000000

0001 - 00011101

0010 - 00100111

0011 - 00111010

0100 - 01001110

0101 - 01010011

0110 - 01101001

0111 - 01110100

1000 - 10000111

1001 - 10011010

1010 - 10100000

1011 - 10111101

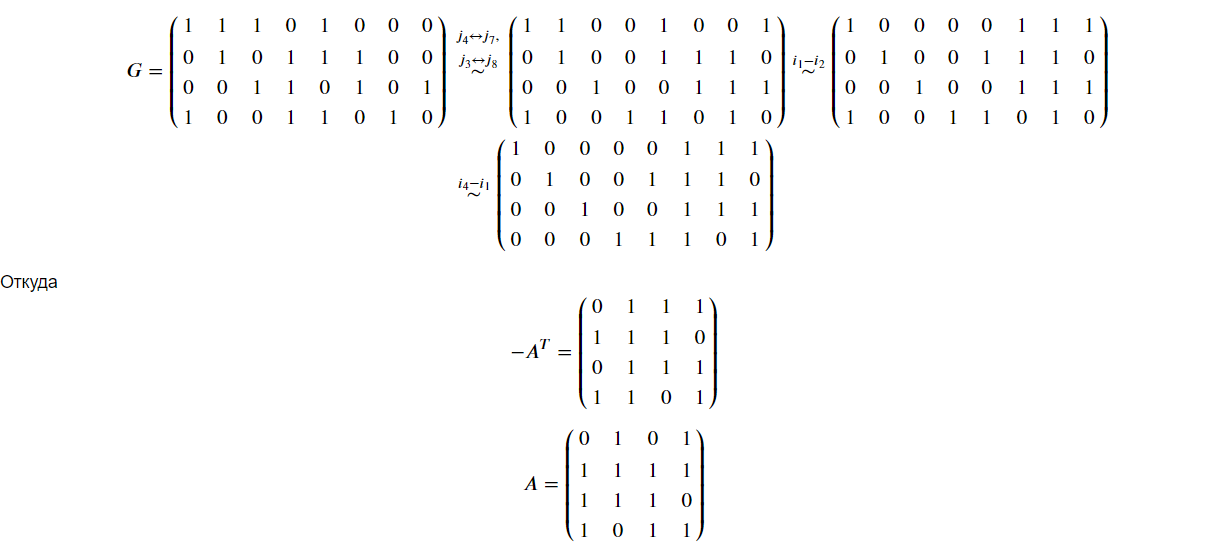
1100 - 11001001

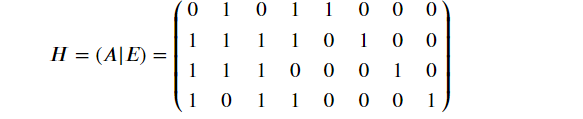
1101 - 11010100

1110 - 11101110

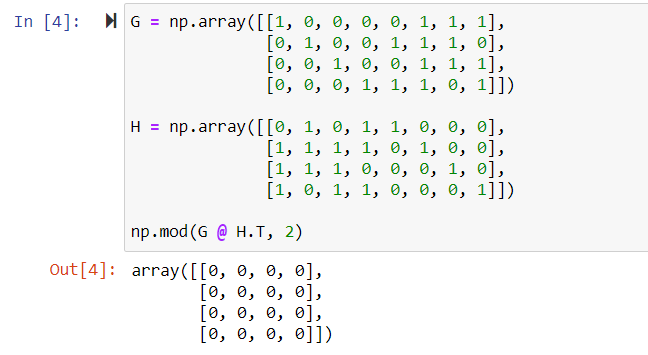
1111 - 11110011

Найдем кодовое расстояние как минимальный вес кодовых слов, не включая первого, содержащего только 0. Минимальный вес равен 2 для кодового слова, соответствующего сообщению 1010.

Задание 1б)  


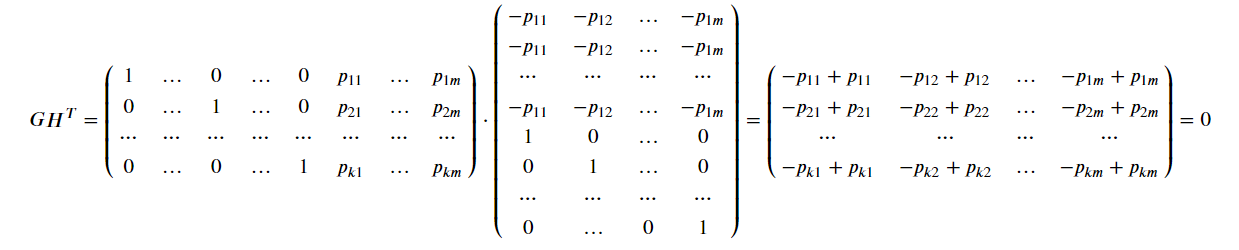


Покажем, что 𝐻 действительно является проверочной для 𝐺 , для этого перемножим 𝐺 (преобразованную) и 𝐻𝑇 :



Задание 2

Покажем, что в качестве проверочной матрицы для 𝐺 можно взять 𝐻, перемножив 𝐺 и 𝐻𝑇:



Произведение 𝐺𝐻𝑇 есть нулевая матрица, следовательно в качестве проверочной матрицы для 𝐺 можно взять 𝐻.

**3. Код программы (к заданию 1а)**

# получение всех сообщений указанной длины

def all\_messages(length):

words =[i for i in range(0, 2\*\*length)]

strings =["{0:b}".format(i).zfill(length) for i in words]

bits = [np.array(list(map(int, str))) for str in strings]

return bits

# получение всех кодовых слов для данной порождающей матрицы

def all\_codewords(G):

length = G.shape[0]

messages = all\_messages(length)

words = [np.mod(x @ G, 2) for x in messages]

return words

# определение веса как количества единиц в кодовом слове

def code\_weight(word):

return np.count\_nonzero(word)

# определение кодового расстояния

def code\_distance(words):

return min(map(code\_weight, words[1:]))

**4. Вывод**

В данной ЛР была написана программа, которая создает двоичную кодовую таблицу из порождающей матрицы и находит ее кодовое расстояние как минимальный вес кодовых слов. Также был показан способ нахождения проверочной матрицы при данной порождающей. Использование порождающей матрицы позволяет ускорить кодирование и, в случае больших кодовых таблиц, экономить память.