**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«Сибирский государственный университет науки и технологий   
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедра информационно-управляющих систем

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Теория информации

|  |
| --- |
| Линейные групповые коды |

Руководитель А.Н. Бочаров

подпись, дата инициалы, фамилия

Обучающийся БПИ22-02, 221219040 К.В.Трифонов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2024 г.

# ЦЕЛЬ РАБОТЫ

Закрепление знаний по методам кодирования информации.

# порядок выполнения работы

1. Ознакомиться с общей постановкой задачи.
2. Ознакомится с вариантом задания – соответствует вашему номеру в списке группы.
3. Выполнить задания согласно варианту.
4. Подготовить отчет по лабораторной работе. Отчет должен включать в себя:   
   •  титульный лист;   
   •  цель лабораторной работы;   
   •  постановку задачи;   
   •  ход работы;   
   •  краткие ответы на контрольные вопросы;   
   •  выводы по лабораторной работе.
5. Защитить лабораторную работу перед преподавателем.

# задания

1. Построить линейный групповой код, способный исправлять одиночную ошибку. Вариант взять аналогичный из лабораторной работы № 4.   
 2. Привести пример построения 10 кодовых комбинаций.   
 3. Показать процесс исправления ошибки в заданном разряде.

4. Составить программу, кодирующую и декодирующую кодовую комбинацию, с целью обнаружения и исправления одиночной ошибки.

Вариант 8. Кол-во сообщений (n) - 2048.

**Вариант 2**:

кол-во сообщений *N* = 32.

# ХОД РАБОТЫ

Кол-во сообщений(n) = 2048 = 211 => nи = 11

nк = = =

= = 3,962 бит ≈ 4 бит

n = nк + nи = 4 + 11 = 15

Разрядность информационной части кода:

*n*и = log2 *N* = log2 32 = 5 бит.

Разрядность контрольной части кода:

*n*к = log2 ((*n*и + 1) + log2 (*n*и + 1)) = log2 ((5 + 1) + log2 (5 + 1)) ≈ 3,1 = 4 бита.

Общая разрядность кода:

*n* = *n*и + *n*к = 5 + 4 = 9 бит.

Wп ≥ dmin – 1 => Wп ≥ 3 – 1 => Wп ≥ 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | **1** | **1** | **1** | **0** |
| 0 | 0 | 1 | 0 | 0 | **1** | **1** | **0** | **1** |
| 0 | 0 | 0 | 1 | 0 | **1** | **1** | **0** | **0** |
| 0 | 0 | 0 | 0 | 1 | **1** | **0** | **1** | **1** |

G9,5 =

Возьмем следующую комбинацию.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a1 | a2 | a3 | a4 | a5 |
| 1 | 1 | 0 | 0 | 1 |

Складываем строки проверочной части, порядковый номер которых совпадает с индексом единиц в нашей взятой строке.

1111 + 1110 + 1011 = 1010 => p1 = 1 p2= 0 p3= 1 p4= 0

Таким образом, итоговый код будет выглядеть следующим образом.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 | a2 | a3 | a4 | a5 | p1 | p2 | p3 | p4 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | **0** | **1** | **0** | **0** |
| 1 | 1 | 0 | 0 | 1 | **0** | **0** | **1** | **0** |
| 1 | 0 | 1 | 0 | 1 | **0** | **0** | **0** | **1** |

H = ПT \* **I**nк =

S1 = p1 + a1 + a2 + a3 + a4 + a5

S2 = p2 + a1 + a2 + a3 + a4

S3 = p3 + a1 + a2 + a5

S4 = p4 + a1 + a3 + a5

Пусть ошибка в 1-м разряде. Тогда код будет выглядеть следующим образом.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a1** | a2 | a3 | a4 | a5 | p1 | p2 | p3 | p4 |
| **0** | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |

S1 = p1 + a1 + a2 + a3 + a4 + a5 = 1 + **0** + 1 + 0 + 0 + 1 = 1

S2 = p2 + a1 + a2 + a3 + a4 = 0 + **0** + 1 + 0 + 0 = 1

S3 = p3 + a1 + a2 + a5 = 1 + **0** + 1 + 1= 1

S4 = p4 + a1 + a3 + a5 = 0 + **0** + 0 + 1 = 1

S

Это соответствует 1-му столбцу матрицы H => ошибка в 1-м разряде.

Далее была разработана программа, которая строит линейно-групповые кода и выполняет обнаружение и исправление ошибок. С помощью нее было сгенерировано 10 кодов согласно заданию.

Демонстрация работы программы (Рисунок 1).

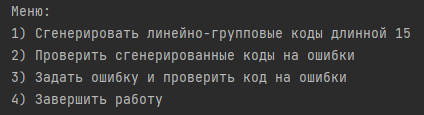


Рисунок 1 – Меню программы

Нажимая действие 1, генерируем 10 кодов (Рисунок 2).

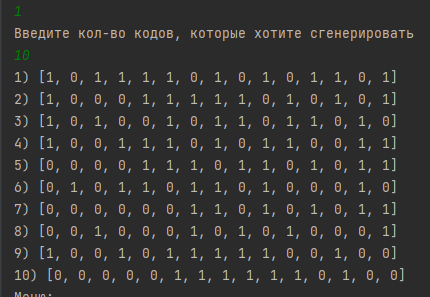


Рисунок 2 – Генерирование десяти кодов

Действием 2 можно проверить сгенерированные коды на наличие ошибок (Рисунок 3).

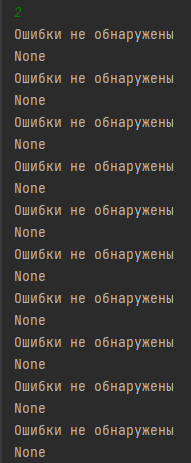


Рисунок 3 – Проверка кодов

Действием 3 можно выбрать код, в котором хотим сделать ошибку, задать индекс для добавления ошибки, далее программа обнаружит и исправит данную ошибку (Рисунок 4).

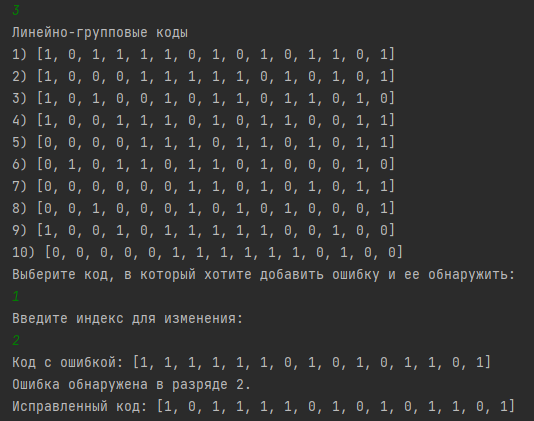


Рисунок 4 – Обнаружение и исправление ошибки

Действием 4 программа завершает свое выполнение (Рисунок 5).

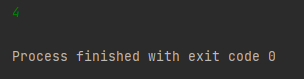


Рисунок 5 – Завершение работы

Листинг программы:

from random import randint  
import numpy as np  
  
matrix\_1=[[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],  
 ]  
  
matrix\_p=[[1,1,1,1],  
 [1,1,1,0],  
 [1,1,0,1],  
 [1,1,0,0],  
 [1,0,1,1],  
 [1,0,1,0],  
 [1,0,0,1],  
 [0,1,1,1],  
 [0,1,1,0],  
 [0,1,0,1],  
 [0,0,1,1]]  
  
matrix\_h=[[1,1,1,1,1,1,1,0,0,0,0],  
 [1,1,1,1,0,0,0,1,1,1,0],  
 [1,1,0,0,1,1,0,1,1,0,1],  
 [1,0,1,0,1,0,1,1,0,1,1]]  
  
matrix\_h\_1=[[1,0,0,0],  
 [0,1,0,0],  
 [0,0,1,0],  
 [0,0,0,1]]  
  
matrix\_h\_for\_decode=[[1,1,1,1],  
 [1,1,1,0],  
 [1,1,0,1],  
 [1,1,0,0],  
 [1,0,1,1],  
 [1,0,1,0],  
 [1,0,0,1],  
 [0,1,1,1],  
 [0,1,1,0],  
 [0,1,0,1],  
 [0,0,1,1],  
 [1,0,0,0],  
 [0,1,0,0],  
 [0,0,1,0],  
 [0,0,0,1]]  
  
def sum\_vectors(vec1,vec2):  
 sum=[]  
 for i in range(len(vec1)):  
 sum.append((vec1[i]+vec2[i])%2)  
 return sum  
  
def get\_codes\_with\_p(count):  
 codes = []  
 for i in range(0, count):  
 random\_codes\_without\_k = [randint(0, 1) for i in range(11)]  
 codes.append(random\_codes\_without\_k)  
 vector\_0=[0,0,0,0]  
 for j in range (len(codes[i])):  
 if codes[i][j] ==1:  
 vector\_0=sum\_vectors(vector\_0,matrix\_p[j])  
 for p in vector\_0:  
 codes[i].append(p)  
 return codes  
  
def decode(code):  
 s=[]  
 s1 = (code[11] + code[0] + code[1] + code[2] + code[3] + code[4] + code[5] + code[6])%2  
 s2 = (code[12] + code[0] + code[1] + code[2] + code[3] + code[7] + code[8] + code[9])%2  
 s3 = (code[13] + code[0] + code[1] + code[4] + code[5] + code[7] + code[8] + code[10])%2  
 s4 = (code[14] + code[0] + code[2] + code[4] + code[6] + code[7] + code[9] + code[10])%2  
 s.append(s1)  
 s.append(s2)  
 s.append(s3)  
 s.append(s4)  
 if s==[0,0,0,0]:  
 print("Ошибки не обнаружены")  
 else:  
 for i in range(len(matrix\_h\_for\_decode)):  
 if matrix\_h\_for\_decode[i]==s:  
 print(f"Ошибка обнаружена в разряде {i+1}.")  
 if code[i] == 0:  
 code[i] = 1  
 else:  
 code[i] = 0  
 print(f"Исправленный код: {code}")  
  
choice=0  
codes=[]  
while choice!=4:  
 print("Меню:")  
 print("1) Сгенерировать линейно-групповые коды длинной 15")  
 print("2) Проверить сгенерированные коды на ошибки")  
 print("3) Задать ошибку и проверить код на ошибки")  
 print("4) Завершить работу")  
 choice = int(input())  
 if choice == 1:  
 print("Введите кол-во кодов, которые хотите сгенерировать")  
 count=int(input())  
 codes = get\_codes\_with\_p(count)  
 for i in range(len(codes)):  
 print(f"{i + 1}) {codes[i]}")  
 if choice == 2:  
 for i in range(len(codes)):  
 print(decode(codes[i]))  
 if choice == 3:  
 print("Линейно-групповые коды")  
 for i in range(len(codes)):  
 print(f"{i + 1}) {codes[i]}")  
 print("Выберите код, в который хотите добавить ошибку и ее обнаружить:")  
 find\_error = int(input())  
 print("Введите индекс для изменения:")  
 index\_for\_create\_error = int(input())  
 if codes[find\_error - 1][index\_for\_create\_error - 1] == 0:  
 codes[find\_error - 1][index\_for\_create\_error - 1] = 1  
 else:  
 codes[find\_error - 1][index\_for\_create\_error - 1] = 0  
 print(f"Код с ошибкой: {codes[find\_error - 1]}")  
 decode(codes[find\_error - 1])

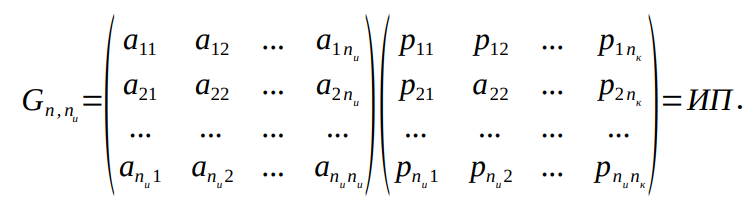
# ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое линейные групповые коды?

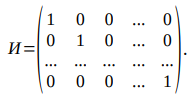
Линейными называют коды, в которых проверочные символы представляют собой линейные комбинации информационных символов.

2. Как строится производящая матрица G?

ЛГК задают матрицами, размеры которых определяются числом информационных и контрольных разрядов. Такие матрицы называются производящими или порождающими. Производящая матрица состоит из двух частей: информационной и проверочной.



В качестве информационной части И выбирается единичная матрица в канонической форме

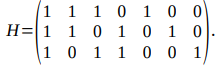


При построении проверочной части П придерживаются следующего правила. Вес каждой строки матрицы П должен быть не менее



3. Как строиться контрольная матрица H?

Строим контрольную матрицу Н. Транспонируем часть П производящей матрицы и дополняем ее единичной матрицей.



4. Как происходить декодирование линейного группового кода?

В процессе декодирования вычисляются схемы проверок, число которых равно числу контрольных разрядов. В результате образуется вектор ошибки - синдром. Если синдром равен нулю, то комбинация принята правильно. Если синдром не нулевой, то имеется ошибка.

5. Что такое систематический код?

Систематические коды — это коды в которых информационные и корректирующие разряды размещаются по определенной системе и всегда занимают строго определенные места в кодовых комбинациях.

6. В чем суть методики построения систематического кода?

Построение систематического кода производится следующим образом:   
 1) В качестве первого вектора берется нулевой вектор.

2) Составляется производящая матрица G, имеющая nи строк и n столбцов. В качестве строк берутся любые ненулевые линейно-независимые n значные векторы, отстоящие друг от друга не менее, чем на заданное кодовое расстояние.

3) Остальные кодовые вектора числом 2 nи−nи−1 получаются как линейные комбинации векторов (в различных сочетаниях), входящих в производящую матрицу.

4) Для обнаружения ошибок в кодовых комбинациях составляется множество проверочных векторов, ортогональных векторам кода. Из проверочных векторов составляют проверочную матрицу H. Данная матрица имеет nк строк и n столбцов. Её строками являются любые линейнонезависимые комбинации проверочных векторов.

7. Какие методы декодирования систематического кода вы знаете?

1) Составление полной кодовой таблицы

2) Декодирование проверкой на чётность

3) Декодирование с помощью корректирующего вектора

8. Как проводится декодирование систематического кода по методу полной кодовой таблицы?

Общий метод декодирования состоит в том, что приняв некий вектор Vx его отыскиваю в полной кодовой таблице и соотносят с тем кодовым вектором, который стоит в верхней строке столбца. Операция исправления заключается в



Данный метод является универсальным, но громоздким.

9. Как проводится декодирование систематического кода по методу проверки на чётность?

На основе условия ортогональности проверочных векторов и векторов кода можно составить правила проверки. На основании правил проверки формируется вектор ошибки (синдром). Если вектор равен нулю, то кодовая комбинация не содержит ошибки. Если вектор ошибки не нулевой, то кодовая комбинация с ошибкой. Исходя из формированных правил проверки и минимального кодового расстояния по вектору ошибки можно определить ошибочный разряд или просто констатировать, что имеется ошибка.

10. Как проводится декодирование систематического кода по методу полного корректирующего вектора?

Исходя из выше изложенного метод декодирования состоит из следующих операций:

1) Вычисляются корректирующие вектора для векторов ошибки.

2) Для принятого вектора Vx определяется корректирующий вектор.

3) По найденному корректирующему вектору определяем вектор ошибки. 4) Ошибку исправляем путём сложения Vx и вектора ошибки.

# ВЫВОДЫ

Таким образом, были получены навыки построения линейно-групповых кодов, обнаружения и исправления одной в ошибки в коде.