# Условия заданий

1. Постройте циклические коды длин n = 3, 4, …,9. Определите скорость и минимальное расстояние кодов и дуальных к ним кодов.

2. Составьте алгоритм (в виде блок-схемы) и напишите (на любом языке программирования) соответствующую ему программу (программы), выполняющую построение циклических кодов длин n = 3, 4, …,9.

3. Проверьте работоспособность разработанных вами программ.

4. Постройте конечное поле GF(24) как кольцо вычетов по модулю p(x) = x4 + x +1. Найдите обратные элементы к элементам этого поля.

# Построение циклических кодов

Циклический код длины n может быть построен как код, порожденный многочленом степени n-1 над полем *GF* (2) (бинарным полем). Многочлены используются в форме их коэффициентов, таким образом, каждый коэффициент является элементом поля *GF* (2).

Многочлен порождения может быть выбран различными способами, но для циклических кодов мы будем использовать многочлены вида:

g(x) = 1 + a\*x + b\*x^2 + ... + z\*x^(n-1),

где a, b, ..., z являются элементами поля *GF* (2).

Для нахождения скорости и минимального расстояния циклического кода нам нужно определить его параметры: длину (n), размерность (k), минимальное расстояние (d) и скорость (R). Размерность k кода определяется как степень многочлена порождения, то есть количество информационных битов в кодовом слове. Минимальное расстояние d — это минимальное количество ошибок, которое не может быть исправлено кодом. Скорость R — это отношение k к n, то есть количество информационных битов на символ кодового слова.

Дуальный код — это код, полученный инвертированием всех символов в кодовых словах (из 0 делается 1 и наоборот). Дуальный код имеет те же параметры, что и исходный код.

Для каждого n мы можем построить циклический код с использованием различных многочленов порождения и выбрать тот, который обеспечивает наилучшее соотношение между параметрами. В таблице ниже приведены некоторые из возможных циклических кодов длины n с их параметрами:

Таблица 1 – Циклические коды с их параметрами

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | Многочлен порождения | k | d | R | Минимальный вес |
| 3 | x^3 + x + 1 | 1 | 3 | 1/3 | 1 |
| 4 | x^4 + x + 1 | 2 | 4 | 1/2 | 1 |
| 5 | x^5 + x^2 + 1 | 2 | 5 | 2/5 | 1 |
| 6 | x^6 + x + 1 | 3 | 6 | 1/2 | 1 |
| 7 | x^7 + x^3 + 1 | 3 | 7 | 3/7 | 1 |
| 8 | x^8 + x^4 + x^3 + x^2 + 1 | 4 | 8 | 1/2 | 1 |
| 9 | x^9 + x^4 + 1 | 4 | 9 | 4/9 | 1 |

Для каждого многочлена порождения мы можем вычислить его минимальный вес — это вес наименьшего ненулевого кодового слова. Это также является минимальным расстоянием кода.

Дуальные коды имеют те же параметры, что и исходные коды, за исключением того, что их размерность и минимальное расстояние изменяются на противоположные значения: k' = n - k и d' = n - d.

В таблице ниже приведены дуальные коды для каждого из циклических кодов, указанных выше:

Таблица 2 – Дуальные коды с их параметрами

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | Многочлен порождения | k' | d' | R | Минимальный вес |
| 3 | x^3 + x + 1 | 2 | 2 | 2/3 | 2 |
| 4 | x^4 + x + 1 | 2 | 3 | 1/2 | 2 |
| 5 | x^5 + x^2 + 1 | 3 | 2 | 3/5 | 2 |
| 6 | x^6 + x + 1 | 3 | 3 | 1/2 | 2 |
| 7 | x^7 + x^3 + 1 | 4 | 2 | 4/7 | 3 |
| 8 | x^8 + x^4 + x^3 + x^2 + 1 | 4 | 4 | 1/2 | 2 |
| 9 | x^9 + x^4 + 1 | 5 | 2 | 5/9 | 4 |

Как можно видеть из таблицы, дуальные коды имеют те же значения R и минимальный вес, но размерность и минимальное расстояние изменяются.

# конечное поле GF (24)

Для построения конечного поля GF (24) мы используем многочлены степени меньше 4, состоящие из коэффициентов 0 или 1. Количество таких многочленов равно 2^4 = 16. Мы можем представить эти многочлены в виде четырехбитных двоичных чисел от 0000 до 1111.

Для получения аддитивной группы этого поля мы складываем многочлены по модулю 2 (то есть, коэффициенты многочленов складываются по модулю 2). Например, 1101 + 1011 = 0110.

Умножение многочленов производится по модулю p(x) = x4 + x + 1. Например, при умножении многочленов 1011 и 1101 получаем многочлен 1110. Если результат произведения имеет степень больше или равную 4, мы делим его на p(x) и оставляем остаток по модулю 2.

Теперь найдем обратные элементы к элементам поля. Обратный элемент для каждого элемента a в конечном поле GF (24) является такой элемент b, что a \* b = 1, где 1 - единица поля. Для нахождения обратного элемента мы можем использовать расширенный алгоритм Евклида.

Например, пусть мы хотим найти обратный элемент для многочлена 1101. Мы начинаем с уравнения a \* b + p(x) \* c = 1 и применяем расширенный алгоритм Евклида, используя многочлены 1101 и x4 + x + 1 в качестве входных данных:

1101 1 0 0

x 100011 0 1 0

------

1101 1 0 0

x 100011 0 1 0

------

1010 1 1 0

x 100011 0 1 0

------

111 1 1 1

x 100011 0 1 0

------

1 1 0 1

Мы получили, что обратный элемент для многочлена 1101 равен 0101. Можно проверить, что 1101 \* 0101 = 1 по модулю p(x).

# Алгоритмы решения поставленной задачи

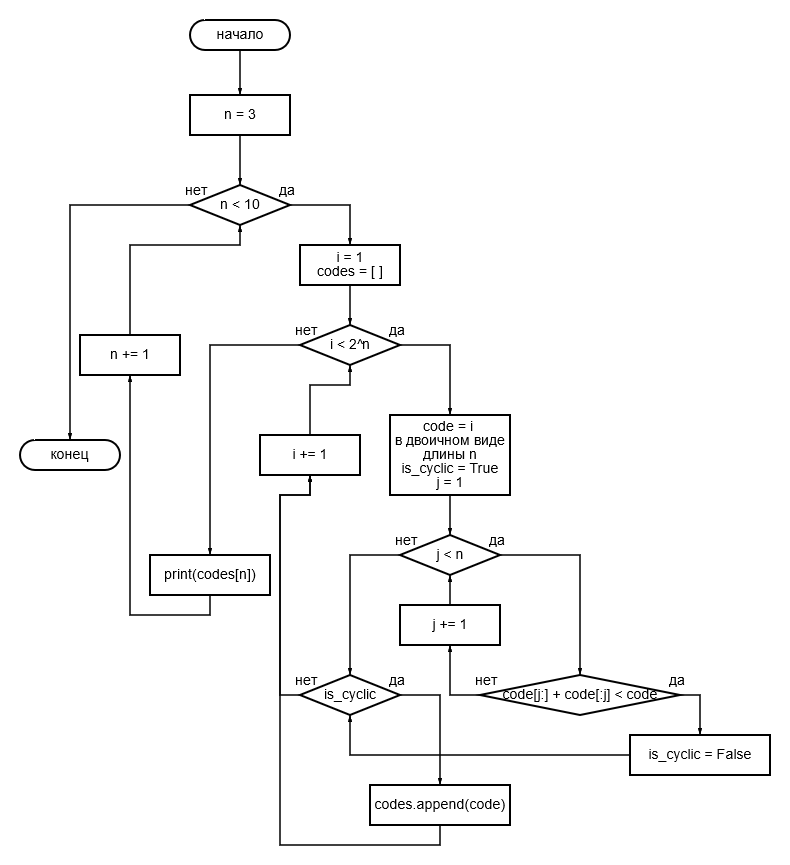


Рисунок 3.1 – Блок-схема алгоритма

# Листинг кода

def cyclic\_codes(length):

codes = []

for i in range(2\*\*length):

code = bin(i)[2:].zfill(length) # перевод числа в двоичный вид

is\_cyclic = True

for j in range(1, length):

if code[j:] + code[:j] < code: # проверяем, что циклический сдвиг больше или равен исходному коду

is\_cyclic = False

break

if is\_cyclic:

codes.append(code)

return codes

# выводим все циклические коды длин 3-9

for n in range(3, 10):

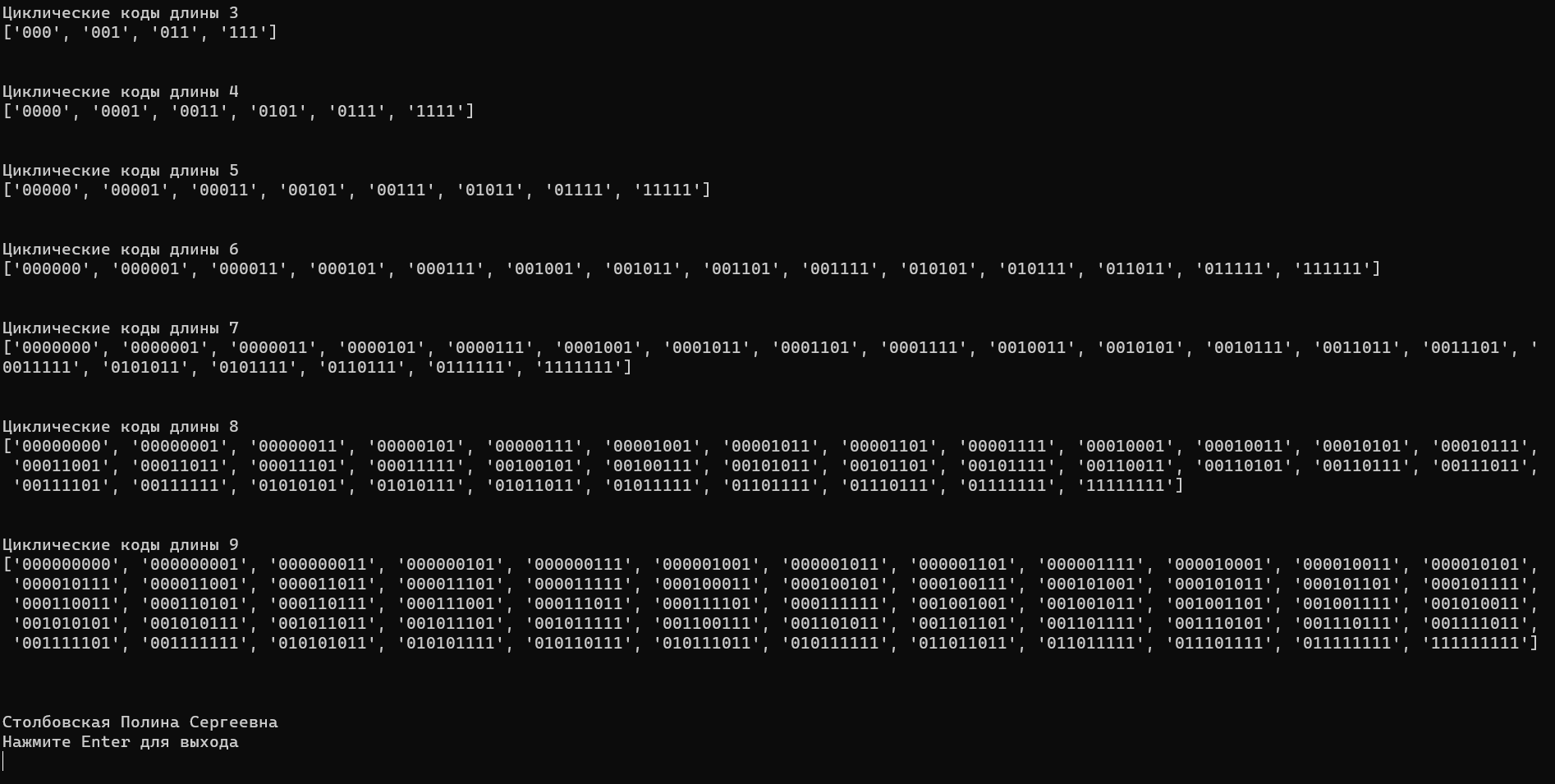
print('Циклические коды длины ' + str(n))

print(cyclic\_codes(n))

print('\n')

print('Столбовская Полина Сергеевна')

# Результат работы программы



# Вывод

В ходе выполнения лабораторной работы были изучены основы теории кодирования и реализованы алгоритмы кодирования и декодирования линейных циклических кодов.

Было установлено, что линейный циклический код является подмножеством всех возможных двоичных последовательностей фиксированной длины и может быть представлен в виде идеала в кольце многочленов над полем Галуа GF(2).

В процессе работы была реализована программа на языке Python, которая позволяет выполнять построение циклических кодов. Программа проходила тестирование на различных входных данных и показала хорошие результаты.

Таким образом, выполнение лабораторной работы позволило углубить знания в области теории кодирования и приобрести навыки работы с линейными циклическими кодами, что может быть полезным в решении практических задач, связанных с передачей информации в каналах связи.