Московский Государственный Университет им. М.В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
Кафедра Суперкомпьютеров и Квантовой Информатики

Курс: Прикладная алгебра.

Отчёт по практическому заданию: Конечные поля и коды БЧХ.

Работу выполнил
студент 323 группы
Тимачев А. А.

Москва 2018

Содержание

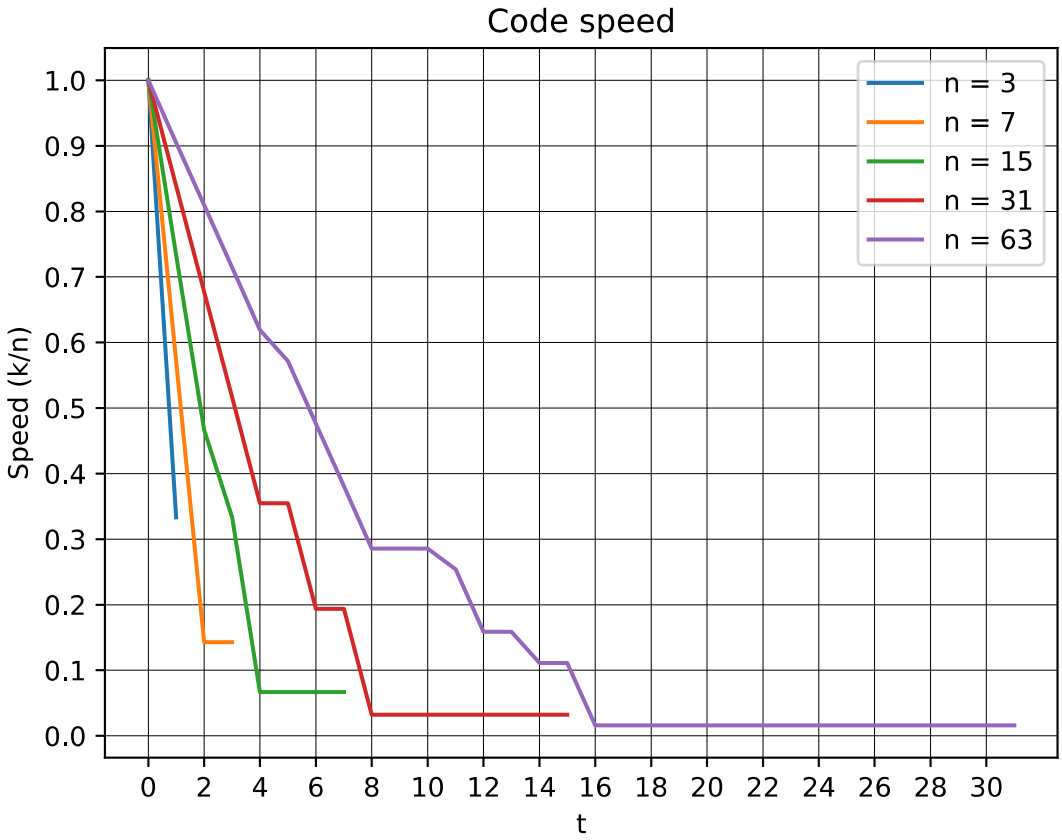
Постановка задачи.....2
Исследование скорости кода БЧХ.....3
Исследование истинного минимального кодового расстояния кода БЧХ.....4
Исследование времени работы декодирования.....5
Исследование эффективности декодирования.....6
Выводы.....9

Постановка задачи.

- Реализовать основные операции в поле $\mathbb{F}_2^q$: сложение, умножение, деление, решение СЛАУ, поиск минимального многочлена из $\mathbb{F}_2[x]$ для заданного набора корней из поля $\mathbb{F}_2^q$;
- Реализовать основные операции для работы с многочленами из $\mathbb{F}_2^q[x]$: произведение многочленов, деление многочленов с остатком, расширенный алгоритм Евклида для пары многочленов, вычисление значения многочлена для набора элементов из $\mathbb{F}_2^q$;
- реализовать процедуру систематического кодирования для циклического кода, заданного своим порождающим многочленом;
- Реализовать процедуру построения порождающего многочлена для БЧХ-кода при заданных n и t;
- Построить графики зависимости скорости БЧХ-кода r = k/n от количества исправляемых кодом ошибок t для различных значений n. Определить какие значения t следует выбирать на практике для заданного n;
- Реализовать процедуру вычисления истинного минимального расстояния циклического кода d, заданного своим порождающим многочленом, путем полного перебора по всем $2^k - 1$ кодовым словам. Привести пример БЧХ-кода, для которого истинное минимальное расстояние больше, чем величина 2t + 1;
- Реализовать процедуру декодирования БЧХ-кода с помощью метода PGZ и на основе расширенного алгоритма Евклида. Провести сравнение двух методов декодирования по времени работы;
- С помощью метода стат. испытаний реализовать процедуру оценки доли правильно раскодированных сообщений, доли ошибочно раскодированных сообщений и доли отказов от декодирования для БЧХ-кода. С помощью этой процедуры убедиться в том, что БЧХ-код действительно позволяет гарантированно исправить до t ошибок, а также необходимо определить может ли БЧХ-код исправить больше, чем t ошибок.

Исследование скорости кода БЧХ.

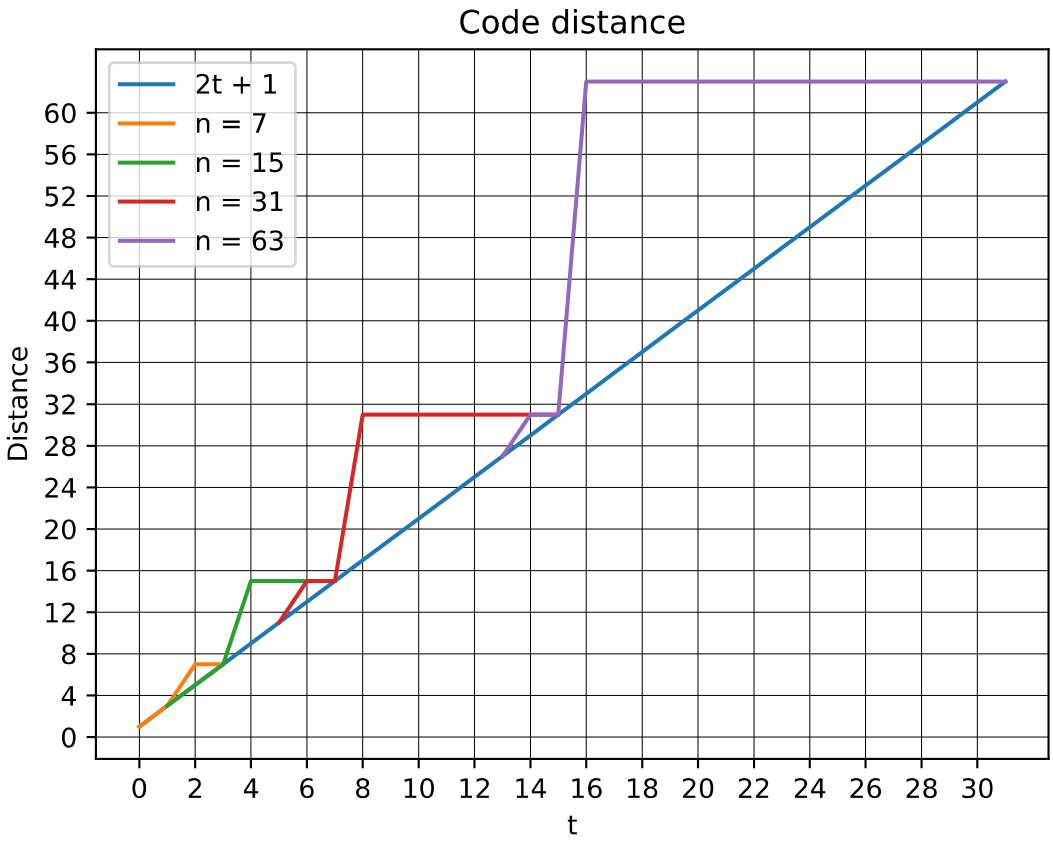
Ниже приведены графики зависимости скорости БЧХ-кодов для разных n от количества исправляемых ими ошибок:



Таким образом, график скорости БЧХ-кода представляет собой невозрастающую функцию. Можно увидеть, что существуют скорости не меняющиеся при небольшом увеличении количества исправляемых ошибок, а это значит, что мы сможем гарантированно исправить большее число ошибок не уменьшив скорость кода. Если мы хотим исправлять больше четверти кодового слова то код будет тривиальным: мы сможем предавать лишь 0 или 1, а кодовое слово будет просто полностью состоять из нулей или единиц. На практике же нужно искать золотую середину между скоростью и числом исправляемых ошибок. Неэффективно брать как слишком маленькое число исправляемых ошибок (скорость будет высокой, однако все сообщения придется очень часто перепосылать), так и слишком медленные (возможно проще иногда лишний раз перепослать, зато с значительно большей скоростью).

Исследование истинного минимального кодового расстояния кода БЧХ.

Ниже приведены минимальные кодовые расстояния для некоторых БЧХ-кодов в виде графиков зависимости расстояния от количества исправляемых ошибок. Количество исправляемых ошибок бралось не с 1, так как при маленьких t и больших n выходит очень большая вычислительная сложность из-за полного перебора. Для наглядности здесь же присутствует прямая 2t + 1.



Из графиков можно увидеть что кодовое расстояние никогда не бывает ниже величины 2t + 1. Причем если для какого-нибудь t расстояние оказалось больше необходимой величины, то при увеличении t оно не будет расти пока обе величины не сравняются. Например, для БЧХ-кода (31, 8) истинное расстояние равно 31, хотя 2t + 1 = 17; для кодов (31, 9), …, (31, 15) кодовое расстояние остается равным 31. Также можно отметить, что эти горизонтальные участки графиков совпадают с постоянством скорости кода.

Исследование времени работы декодирования.

Ниже приведено время работы полного процесса декодирования для набора из 100 кодовых слов. Использовались следующие декодеры: euclid — основанный на расширенном алгоритме Евклида, PGZ (Peterson–Gorenstein–Zierler) — основан на непосредственном решении СЛАУ.

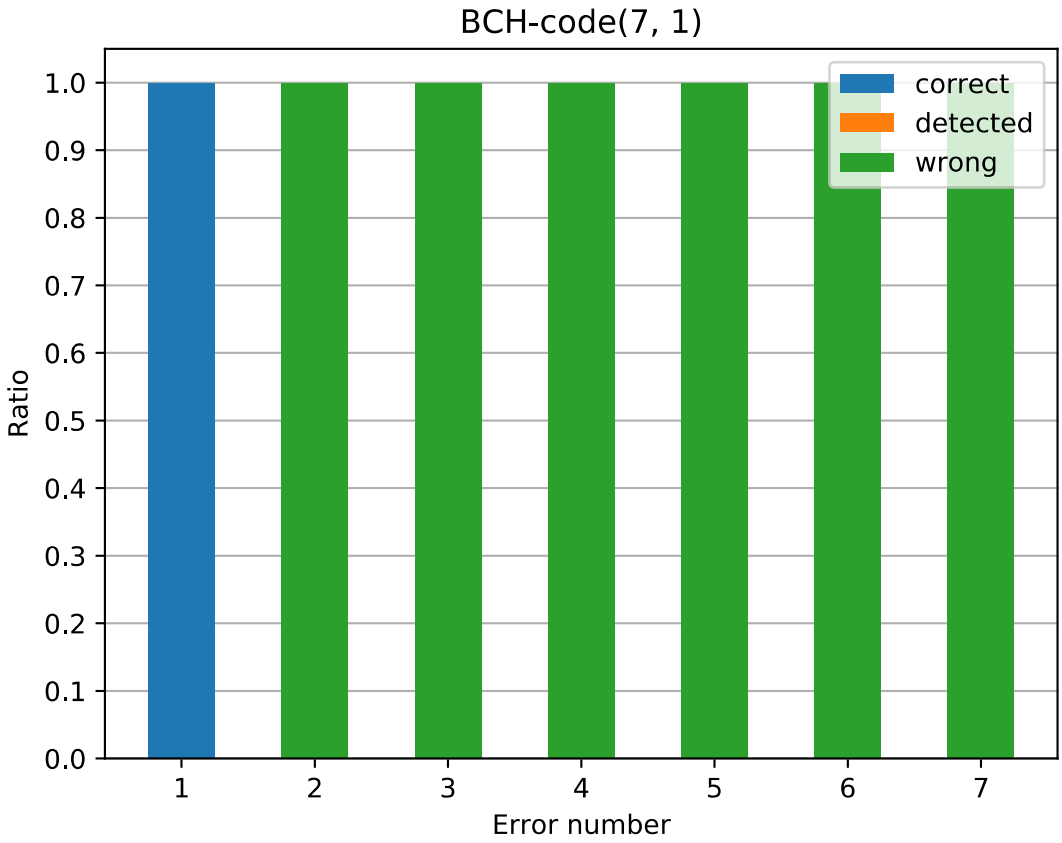
Параметры БЧХ-кода		Совершена 1 ошибка		Совершено t ошибок	
n	t	euclid	PGZ	euclid	PGZ
7	1	0.078001	0.093601	0.078001	0.093601
	2	0.093601	0.140401	0.140401	0.156001
	3	0.124813	0.202801	0.140401	0.156001
15	1	0.140401	0.187201	0.140401	0.187201
	3	0.234002	0.390002	0.280801	0.327602
	7	0.436802	0.951606	0.577203	0.811205
31	1	0.249601	0.358802	0.249601	0.358802
	3	0.468003	0.764404	0.577203	0.780005
	7	0.873605	1.731611	1.232407	1.731611
63	1	0.514803	0.702004	0.514803	0.702004
	5	1.326008	2.371215	1.68481	2.402415
	11	2.542816	5.382034	3.291621	4.77363
127	1	1.029606	1.388408	1.029606	1.388408
	7	3.463222	6.411641	4.851631	7.534848
	15	6.676842	14.289691	10.030864	15.974502

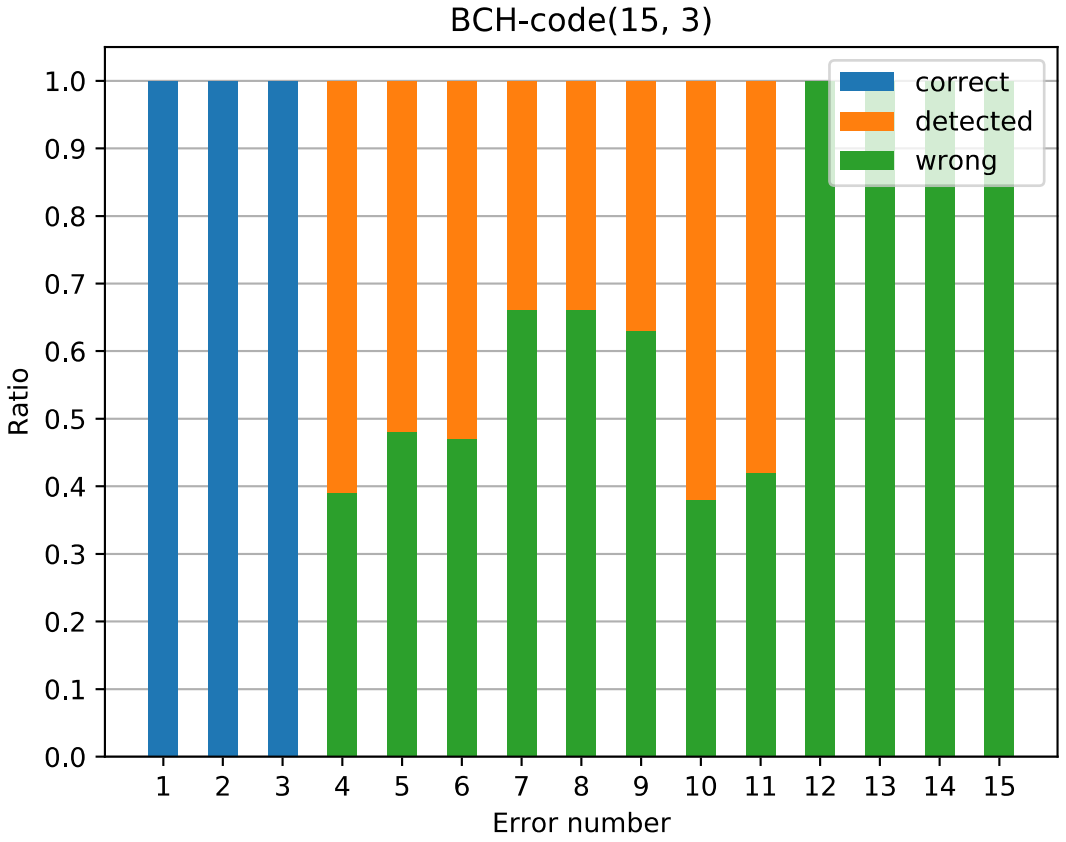
Во всех приведенных случаях euclid показывает лучшее время. При этом если рассматривать код с конкретными параметрами, то при увеличении количества совершенных ошибок время работы euclid однозначно растет, в то время как в PGZ оно может и упасть. Это происходит из-за того, что приходится решать меньшее число СЛАУ. Однако иногда время и увеличивается, скорее всего это происходит из-за того, что при малом числе ошибок СЛАУ не решаются целиком, практически сразу выдается ошибка о том, что матрица в левой части вырождена.

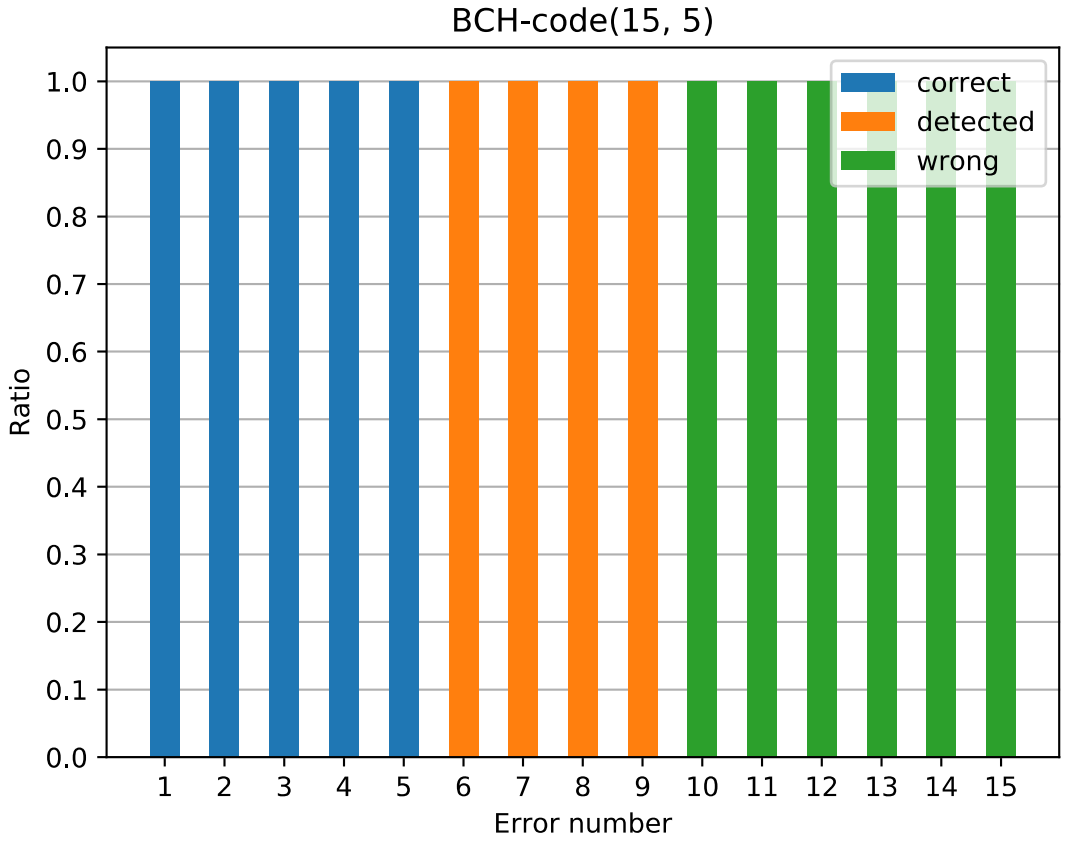
Если же рассматривать одно и то же количество совершаемых ошибок, но разные параметры кода БЧХ, то можно сделать следующий вывод: с увеличением как длины кодовых слов, так и количества исправляемых ошибок время декодирования будет расти для обоих декодеров.

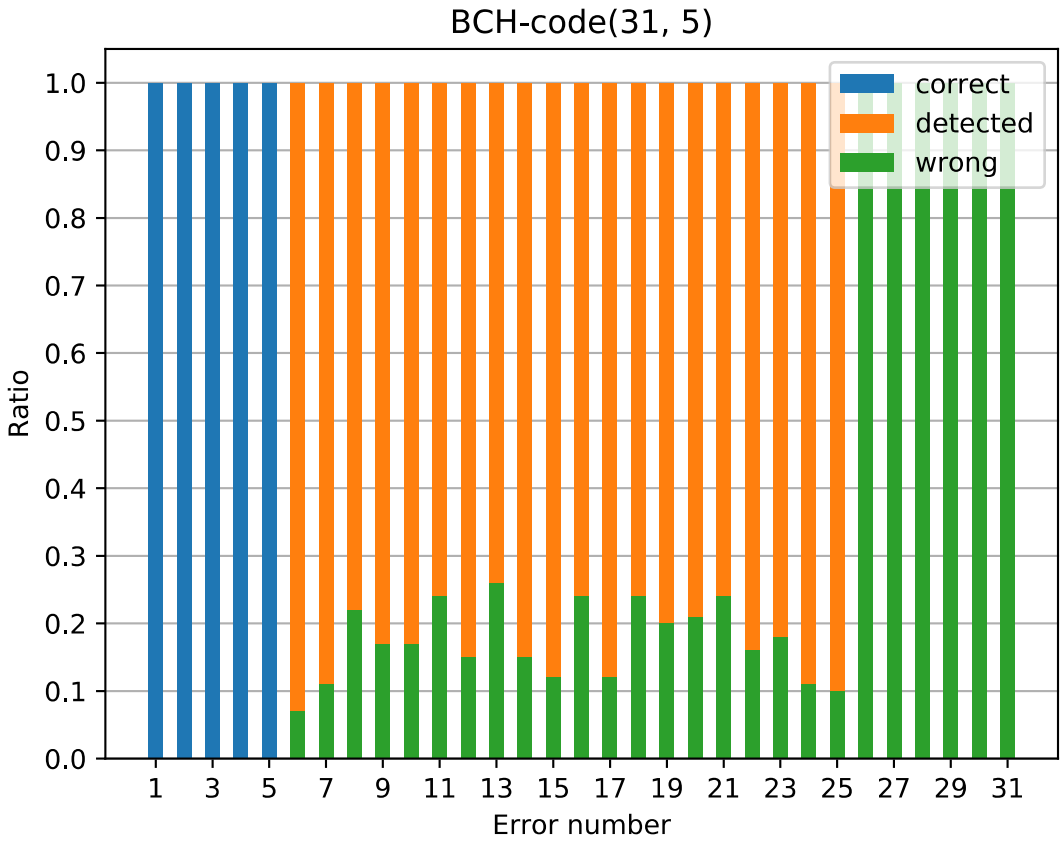
Исследование эффективности декодирования.

Для проверки эффективности декодирования использовался следующий эксперимент: сначала генерировалось 100 случайных сообщений; затем, они кодировались и получались корректные кодовые слова; далее в этих словах менялось конкретное число бит и проводилась попытка их декодирования. Если декодированное сообщение совпадало с исходным, это помечалось как «correct», если декодер выдал отказ от декодирования, то это помечалось как «detected», иначе декодер выдавал сообщение не совпадающее с исходным, и эта ситуация обозначалась как «wrong». Ниже приведены диаграммы с результатами эксперимента для некоторых БЧХ-кодов.




BCH-code(15, 3)
correct
detected
wrong
Ratio
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Error number


BCH-code(15, 5)
correct
detected
wrong
Ratio
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Error number



Из экспериментов можно сделать вывод, что БЧХ-код всегда исправляет до t ошибок включительно. Но вот уже при большем количестве ошибок исходное сообщение восстановить не получается. Причем, иногда получается даже так, что восстанавливается совершенно другое слово. Например, в первом случае (n = 7, t = 1) при любом количестве ошибок кроме 1 будет однозначно декодировано другое слово. Это происходит из-за высокой плотности кода, расстояние между любыми кодовыми словами равно 3, и уже при любых двух ошибках полученное слово оказывается ближе к другому. Существуют также БЧХ-коды которые могут если не исправить, то хотя бы обнаружить больше чем t ошибок. Например, код с параметрами n = 15, t = 5, исправляет до 5 ошибок, но может обнаруживать до 9.

Выводы.

Были реализованы все необходимые модули и операции, а именно:

- Модуль `gf.py` с реализацией основных операций в конечном поле $\mathbb{F}_2^q$ и операций над многочленами из $\mathbb{F}_2^q[x]$;
- Модуль `bch.py` с реализацией БЧХ кодов в виде класса `BCH`.

Также же был написан файл `research.py`, в котором проводились все необходимые исследования. Все заключения по каждому из исследований были написаны в соответствующих разделах данного отчета.

Таким образом, были выполнены все требования практического задания, проведены необходимые исследования и сделаны выводы.