Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**Отчет**

**По домашнему заданию №1**

**По курсу «Сети и телекоммуникации»**

**Вариант 13**

**ИСПОЛНИТЕЛЬ:**

Группа ИУ5-55Б

Обухов А.А.

"12"октября 2023 г.

**ПРЕПОДАВАТЕЛЬ:**

\_\_Галкин В.А.\_\_

" " 2023 г.

Москва 2023

**Постановка и метод решения задачи для варианта задания.**

Имеется дискретный канал связи, на вход которого подается закодированная в соответствии с вариантом задания кодовая последовательность. В канале возможны ошибки любой кратности. Вектор ошибки может принимать значения от единицы в младшем разряде до единицы во всех разрядах кодового вектора. Для каждого значения вектора ошибки на выходе канала после декодирования определяется факт наличия ошибки.

**Вариант № 13**

Требуется, используякодирование циклическим кодом Ц [15,11], определить Корректирующую способность этого кода С0:

|  |  |  |  |
| --- | --- | --- | --- |
| № варианта | Информационный  вектор | Код | Способность  кода |
| 13 | 00001010011 | Ц [15,11] | Ck |

**Обозначения.**

* n – число разрядов в закодированной записи. n = 15.
* k – число разрядов в информационной части. k = 11.
* z – число дополнительных битов для обнаружения ошибки.

.

* r – число разрядов в образующем векторе.
* Ц[15,11] – Циклический код с образующим полиномом.
* Ck - Корректирующая способность кода.
* Деление (в текущей предметной области) – операция О[2].
* g(x) - образующий полином степени z = 4. Для n = 15 он ищется среди простых делителей полинома x15+ 1 [2]. Среди таких есть три полинома четвертой степени, двоичная запись которого содержит 5 бит:

x4 + x3 + x2 + x + 1 => 3110 = 111112

x4 + x3 + 1 => 2510 = 110012

x4 + x + 1 => 1910 = 100112

Возьму в качестве образующего наименьший:

g(x) = x4 + x + 1.

* Vi - информационный вектор. По заданию 00001010011.
* Vc – циклический вектор – результат кодирования Vi.
* I – входящий вектор. Циклический вектор, возможно содержащий ошибки.

**Алгоритмы кодирования, декодирования, вычисления обнаруживающей способности кода для ошибок всех возможных кратностей.**

1. **Алгоритм кодирования циклическим кодом:** 
   1. Осуществить сдвиг информационного вектора на z разрядов влево, заполнив новые младшие разрядов нулями.
   2. Разделить полученный вектор на вектор порождающего полинома g(x), после получить остаток p(x).
   3. Порождающий полином для циклического кода [15,11] имеет вид g(x) = x4 + x + 1.
   4. Применив операцию конкатенации, объединить вектор остатка p(x) и исходный кодовый вектор для получения закодированной последовательности.
2. **Алгоритм декодирования циклическим кодом.**

После передачи кодового полинома V(x) по каналу связи принятый полином r(x) может содержать ошибки. При декодировании циклического кода можно с большой долей вероятности определить, была ли ошибка или нет. Декодирование выполняется следующим образом:

* 1. Разделить принятый полином r(x) на порождающий полином g(x) и проверить остаток от деления S(x) – он является синдромом ошибки.
  2. Если S(x) = 0, то ошибки нет или она не была обнаружена.
  3. Если S(x) ≠ 0, то ошибка есть. По виду вектора синдрома определить место ошибки и исправить ее.
  4. Осуществить сдвиг в сторону младших разрядов исправленного вектора на z разрядов, чтобы получить первоначальный информационный вектор.

1. **Алгоритм вычисления корректирующей способности кода Сk для ошибок всех возможных кратностей**.

Корректирующая способность кода Ck определяется как отношение числа исправленных ошибок Nk к общему числу ошибок данной кратности, которое определяется как число сочетаний из n (длина кодовой комбинации) по i (кратность ошибки – число единиц в векторе ошибок) -

Корректирующая способность кода вычисляется как

Для подсчета корректирующей способности нужно перебрать все возможные вектора ошибок. Их число

Ошибки нужно сгруппировать по кратности. Для каждой группы ввести счетчик , который по умолчанию будет равен 0.

Далее нужно закодировать информационный вектор циклическим алгоритмом, наложив на него вектор ошибки e, а затем, воспользовавшись алгоритмом коррекции, декодировать получившуюся последовательность и сравнить ее с заданным информационным вектором. Если информационный и декодированный вектора равны, то коррекцию ошибки признать успешной и увеличить счетчик на 1, иначе коррекцию ошибки признать неудачной и, не трогая счетчик, продолжить выполнение алгоритма. Текущий пункт повторить раз для каждой ошибки в группе.

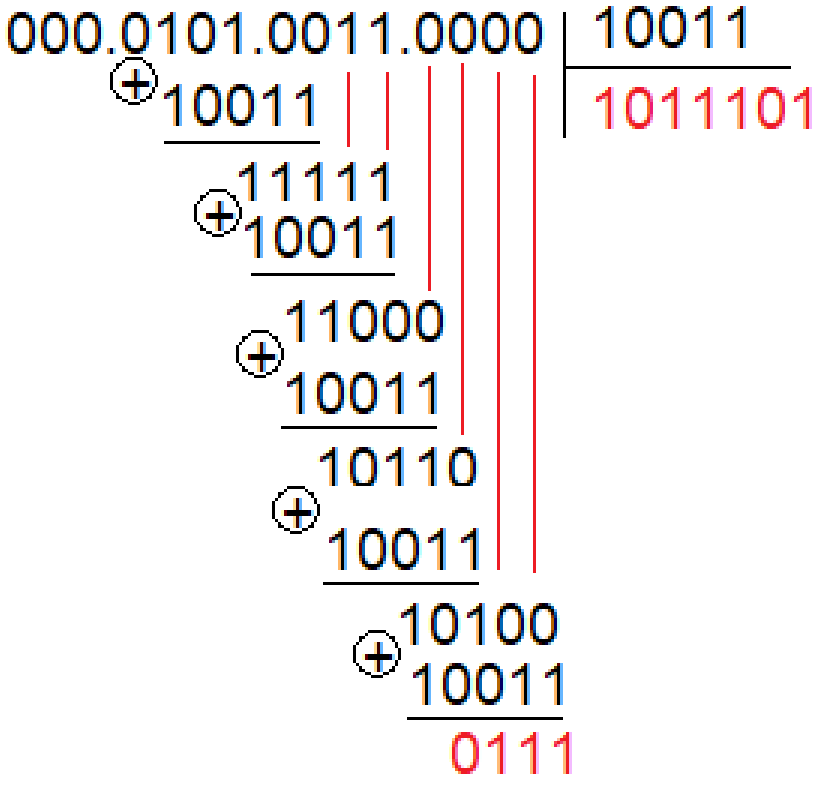
Вышеизложенный алгоритм нужно повторить n раз для ошибок каждой кратности, получив при этом n значений для каждой группы. Составить результирующую таблицу корректирующей способности где номеру группы *i* будет соответствовать кратность ошибки *i* и значение , выраженное в процентах.

1. **Алгоритм коррекции ошибки:**
   1. Составить таблицу соответствия синдрома ошибки Se ошибке e. Для этого для каждой ошибки рассчитать ее синдром Se, сгруппировать синдромы по кратности ошибки.
   2. Найти в таблице синдром-ошибка синдром Sv, полученный при декодировании входной последовательности V, и определить таким образом вектор ошибки.
   3. Инвертировать те разряды во входной последовательности, которые отмечены в векторе ошибки как 1.
   4. Декодировать измененную входную последовательность еще раз.

**Получение циклического кода по варианту задания.**

vi = 000.0101.0011. g(x) = 10011.

1. Сдвиг на 4 бита влево: 000.0101.0011.0000
2. Деление на образующий полином:



1. Конкатенация информационного вектора с остатком: 000.0101.0011.0111

**Вычисление таблицы симптом-ошибка.**

Т.к. для n = 15 число возможных ошибок достигает 2n - 1 = 32767 для этой задачи целесообразно написать компьютерную программу. Для этой цели хорошо подойдет язык golang, т.к. в нем реализованы побитовые операции, такие как ИСКЛЮЧАЮЩЕЕ-ИЛИ (XOR ^), средства визуализации данных с помощью шаблонизатора для HTML-страниц, удобная работа с памятью, массивами, словарями.

Программу можно скачать по ссылке **П.8**

Всего есть 15 классов ошибок: от ошибок разрядности 1 до ошибок разрядности 15. Число ошибок в каждом разряде определяется по формуле: , где *i* – кратность ошибки. Все возможные ошибки, сгруппированные по классам представлены в таблице **П.1**. Число ошибок в каждом классе подсчитано на компьютере и приведено в списке **П.2**, их общее число равно рассчитанному выше значению 32767.

Реализовав алгоритм операции О, нужно перебором ошибок подсчитать для каждой соответствующий симптом. Таблица с расчетами представлена в **П.3**.

Однако при внимательном анализе можно заметить совпадающие симптомы, например, 00102 у ошибок 110.1110.0100.10012 (число 28233) и 0110.1110.0101.10102 (число 28250). И это неудивительно, ведь вектор симптома имеет слишком мало бит, чтобы покрыть все 32767 ошибок, и с образующим полиномом 100112 (19) может предоставить только 18 уникальных ненулевых значений. Кроме того, в таблице наблюдаются множество ошибок, имеющих симптом 0 (**П.4**).

Попытка сгруппировать ошибки по симптомам, чтобы затем по значению симптома можно было попытаться исправить ошибку в полученной последовательности, естественным образом приводит к таблице из 19 симптомов со значениями от 0 до 18 (**П.5**). Естественно, однозначно определить по ним ошибку нельзя, а перебор всех ошибок с данным симптомом при декодировании ничего не даст, ведь все «исправленные» значения дадут разную информационную последовательность.

Именно в этом заключается отнюдь не стопроцентная корректирующая способность.

В реальном канале связи можно было бы статистическими методами определить наиболее вероятную кратность ошибки.

Поэтому целесообразно составить таблицы симптомов для ошибок заданной кратности. (**П.6-7**). Для кратностей 1, 14 и 15 количества возможных остатков хватает, чтобы покрыть все ошибки уникальным симптомом. Для ошибок других кратностей возникают коллизии.

**Список используемой литературы.**

1. Галкин В.А. Методическое пособие по выполнению домашнего задания по дисциплине «Сети и телекоммуникации», 2018
2. Статья на «Портале сетевых проектов», «Кодирование информации. Циклические коды», 2005-2016 гг. <http://project.net.ru/others/article7/net3_9.html>
3. Научная библиотека scask. К.К. Васильев, В.А. Глушков, А.В. Дормидонтов, А.Г. Нестеренко. «Теория электрической связи: учебное пособие». Ульяновск: УлГТУ, 2008. – 452 с. [214]

<https://scask.ru/a_book_tec.php?id=82>

**Приложение.**

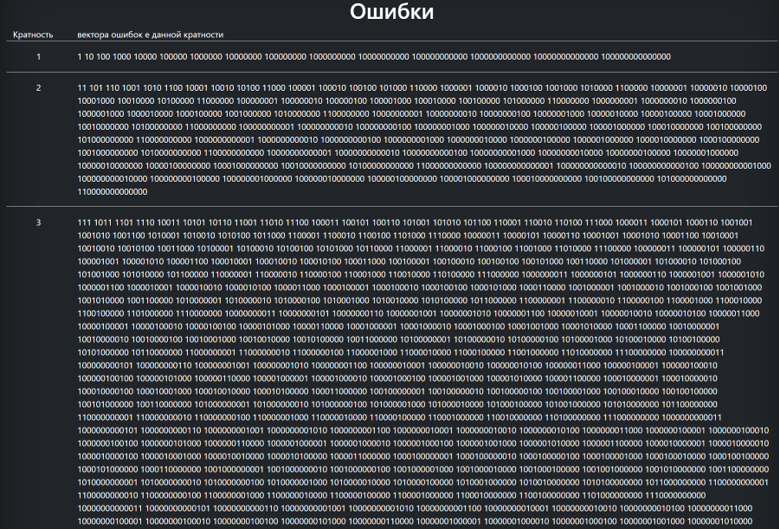


Таблица 1. Фрагмент таблицы разрядность – вектора ошибок данной кратности.

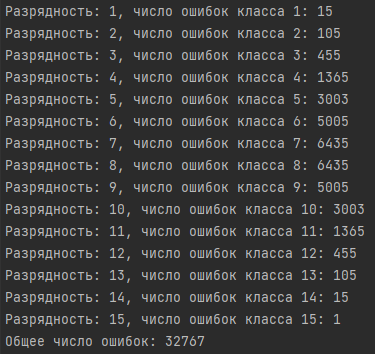
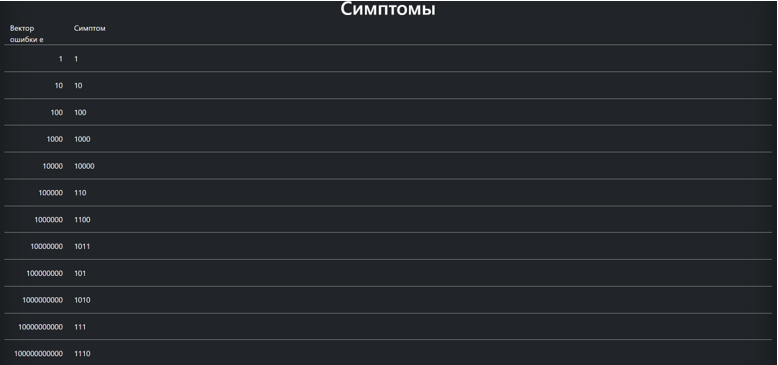
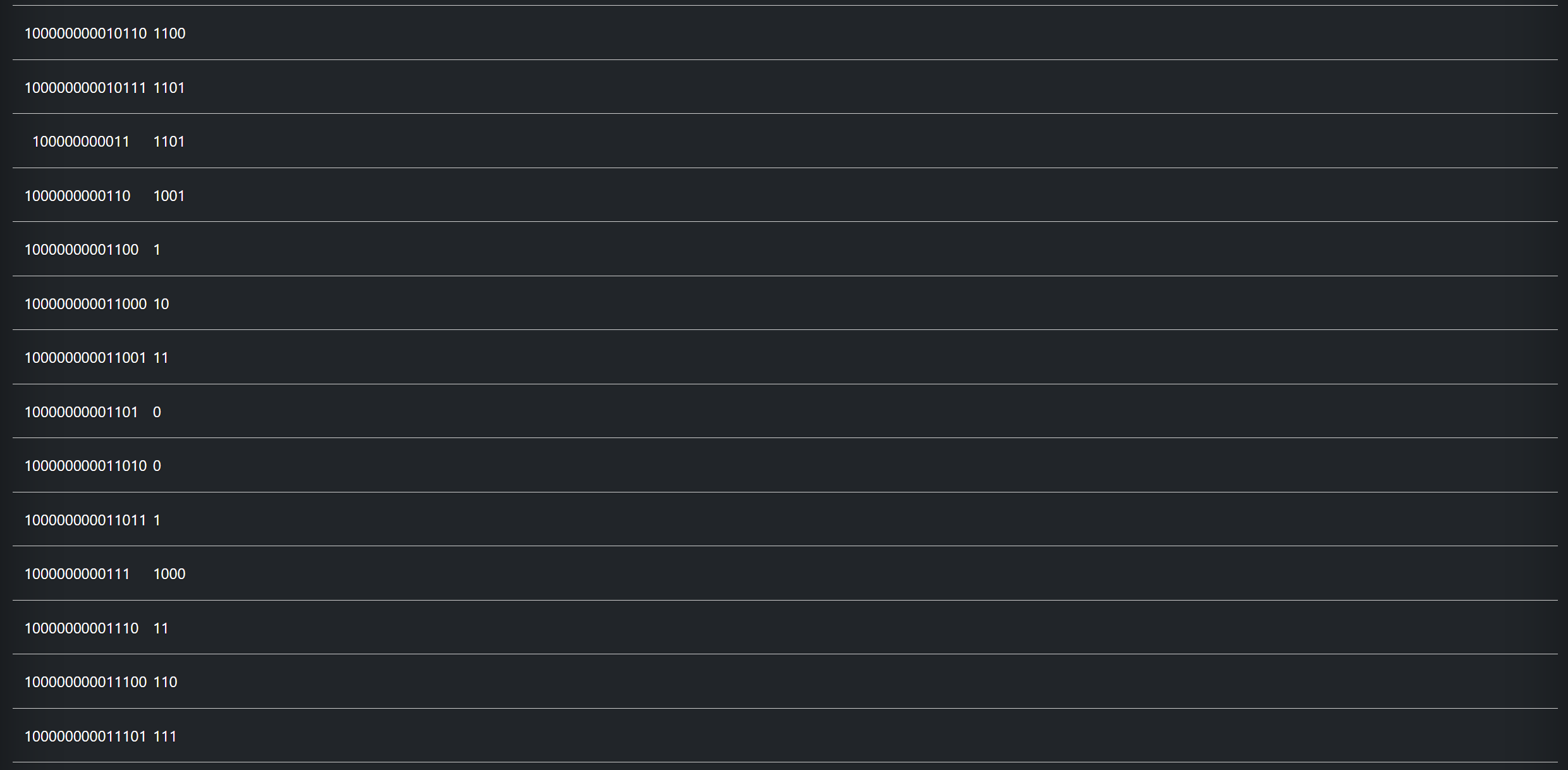


Таблица 2. Подсчет числа ошибок в классе разрядности.



\*\*\*

Фрагменты таблицы 3. Подсчет симптомов к каждой ошибке.



Фрагменты таблицы 4. Совпадающие симптомы в таблице подсчета симптомов к каждой ошибке.

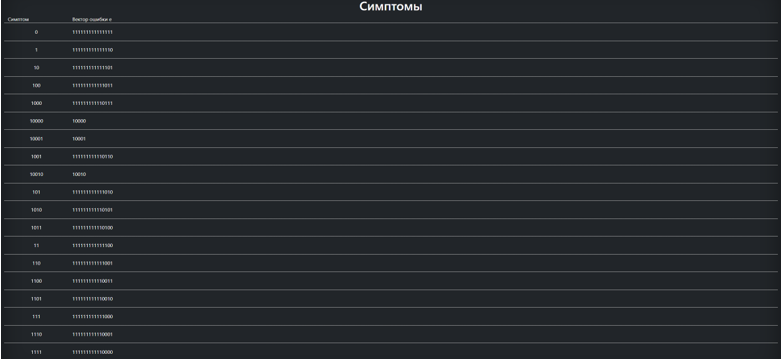


Таблица 5. Симптомы и одна из соответствующих ошибок.

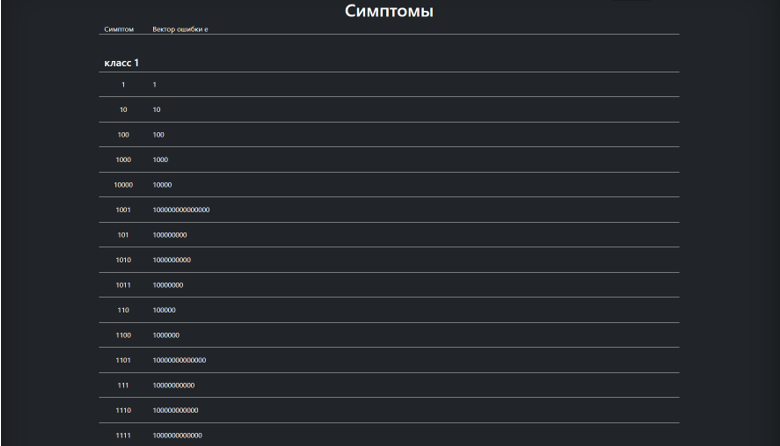


Таблица 6. Симптомы ошибок кратности 1

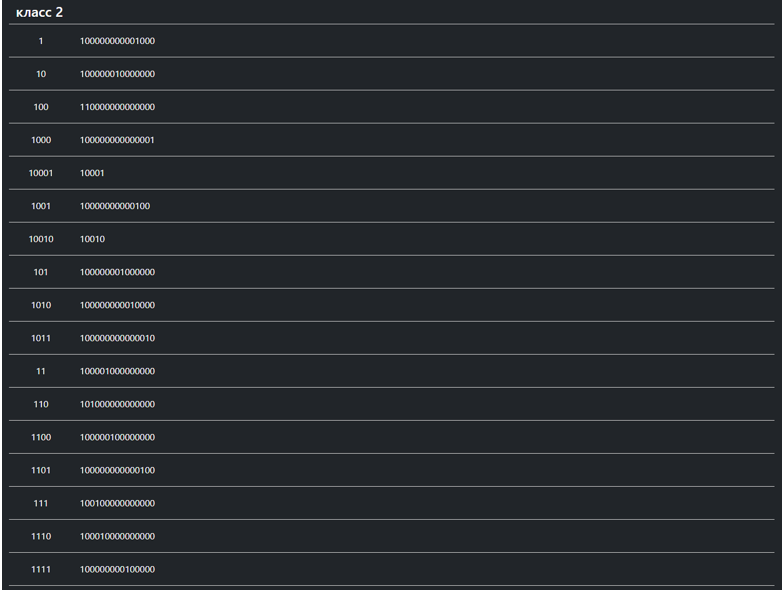


Таблица 7. Симптомы ошибок кратности 2

Программа и отчет 8. https://github.com/Fga4643/Lab\_BMSTU/tree/DZ\_ST/ST%20DZ%201-2%20OBUKHOV