**Лабораторная работа № 6**

**ИЗБЫТОЧНОЕ КОДИРОВАНИЕ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ. ЦИКЛИЧЕСКИЕ КОДЫ**

**Цель:** приобретение практических навыков кодирования/декодирования двоичных данных при использовании циклических кодов (ЦК).

**Теоретические сведения**

**Циклические коды** − это семейство помехоустойчивых кодов, одной из разновидностей которых являются коды Хемминга.

*Основные свойства ЦК:*

• относятся к классу линейных, систематических;

• сумма по модулю 2 двух разрешенных кодовых комбинаций дает также разрешенную кодовую комбинацию;

• каждый вектор (кодовое слово), получаемый из исходного кодового вектора путем циклической перестановки его символов, также является разрешенным кодовым вектором; к примеру, если кодовое слово имеет следующий вид: 1101100, то разрешенной кодовой комбинацией будет и такая: 0110110;

• при простейшей циклической перестановке символы кодового слова перемещаются слева направо на одну позицию, как в приведенном примере;

• поскольку к числу разрешенных кодовых комбинаций ЦК относится нулевая комбинация 000...00, то минимальное кодовое расстояние dmin для ЦК определяется минимальным весом разрешенной кодовой комбинации;

• циклический код не обнаруживает только такие искаженные помехами кодовые комбинации, которые приводят к появлению на стороне приема других разрешенных комбинаций этого кода;

• в основе описания и использования ЦК лежит полином или многочлен некоторой переменной (обычно Х).

Действия с кодовыми словами производятся по **правилам арифметики по модулю 2**. Следует помнить, что **вычитание равносильно сложению.**

Порождающие полиномы циклических кодов.

Характеризуя ЦК в общем случае, обычно отмечают следующее: ЦК составляют множество многочленов {Вj(X)} степени r (r − число проверочных символов в кодовом слове), кратных порождающему (образующему) полиному G(Х) степени r, который должен быть делителем бинома Xn + 1, т. е. остаток после деления бинома на G(X) должен быть нулевым.

Формирование разрешенных кодовых комбинаций ЦК Bj(X) основано на предварительном выборе **порождающего** (генераторного или образующего) полинома G(X), который обладает важным отличительным признаком: **все комбинации Bj(X) делятся на порождающий полином G(X) без остатка.**

Bj(X) / G(X) = Aj(X),

или

Xn / G(X) = Xk

Степень порождающего полинома определяет число проверочных символов: **r = n – k.**

**Порождающими** могут быть только такие полиномы, которые являются делителями двучлена (бинома) x2 + 1. То есть при делении остаток 0.

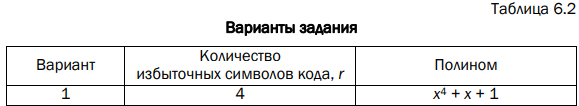


**Синдромом ошибки** в этих кодах является наличие остатка от деления принятой кодовой комбинации на порождающий полином. Если синдром равен нулю, то считается, что ошибок нет.

В противном случае с помощью полученного синдрома можно определить номер разряда принятой кодовой комбинации, в котором произошла ошибка, и исправить ее примерно по той же схеме, которую мы использовали для кода Хемминга.

**Практическое задание**

1. Задание выполняется по указанию преподавателя в соответствии с вариантом из табл. 6.2, из которого выбирается порождающий полином ЦК, а по значению соответствующего ему значения r – длина k информационного слова Xk. Полагаем, что каждый полином соответствует коду, обнаруживающему и исправляющему одиночные ошибки в кодовых словах. Определить параметры (n, k)-кода для своего варианта. Основой задания является разработка приложения.



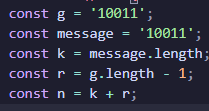


Рисунок 1.1 – Исходные данные и параметры n, k

2. Составить порождающую матрицу (n, k)-кода в соответствии с формулой (6.7), трансформировать ее в каноническую форму и далее – в проверочную матрицу канонической формы.

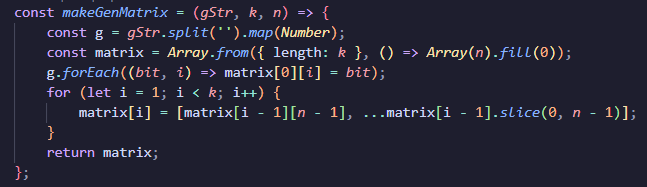


Рисунок 1.2 – Генерация матрицы

3. Используя порождающую матрицу ЦК, вычислить избыточные символы (слово Xr) кодового слова Xn и сформировать это кодовое слово.

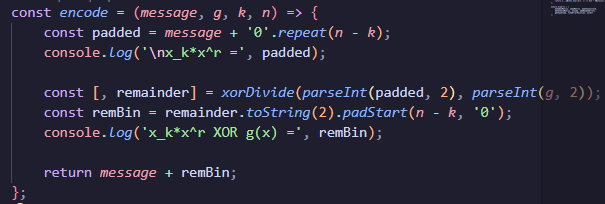


Рисунок 1.3 – Вычисление избыточных символов

4. Принять кодовое слово Yn со следующим числом ошибок: 0; 1; 2. Позиция ошибки определяется (генерируется) случайным образом.

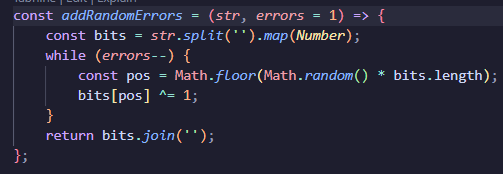


Рисунок 1.4 – Генерация ошибок

5. Для полученного слова Yn вычислить и проанализировать синдром. В случае, если анализ синдрома показал, что информационное сообщение было передано с ошибкой (или 2 ошибками), сгенерировать унарный вектор ошибки Еn = е1, е2, …, еn и исправить одиночную ошибку, используя выражение (6.5); проанализировать ситуацию при возникновении ошибки в 2 битах.

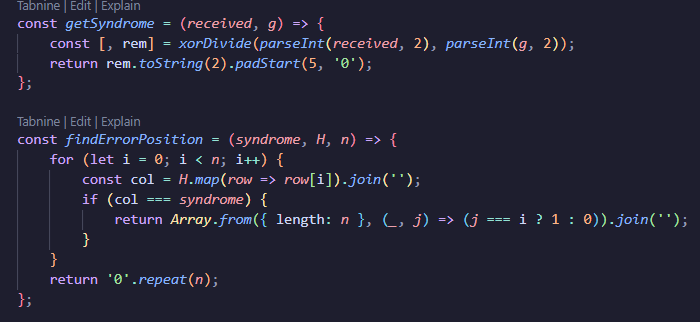


Рисунок 1.5 – Вычисление синдрома и позиций ошибок

6. Результаты оформить в виде отчета по установленным правилам.

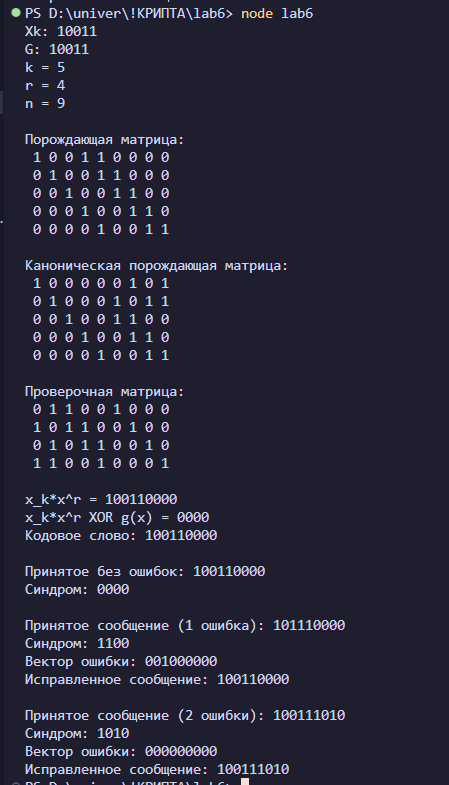
+

Рисунок 1.6 - Результат