**Лабораторная работа № 7**

**ПЕРЕМЕЖЕНИЕ/ДЕПЕРЕМЕЖЕНИЕ ДАННЫХ В ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ**

**Цель:** приобретение практических навыков использования методов перемежения/деперемежения двоичных данных в информационных системах.

**Теоретические сведения**

Довольно часто распределение ошибок носит взаимозависимый характер. В таких случаях говорят о группах (или пакетах) ошибок.

Существуют специальные коды, корректирующие пакетные ошибки, однако на практике чаще используют перемежение/деперемежение совместно с традиционными кодами.

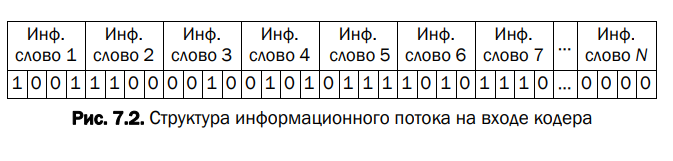
Идея перемежения/деперемежения состоит в следующем. Если биты каждого кодового слова Хn передаются не в обычной последовательности, а через интервалы, превышающие ожидаемую длину пакета ошибок (в промежутки между битами одного слова вставляются биты других кодовых слов), то при возникновении такого типа ошибки обратная перемежению операция – деперемежение – разнесет («размажет») группу ошибок по всей совокупности кодовых слов, составляющих данное сообщение.

Длина пакета в нашем случае – это число рядом расположенных ошибочных битов.

Предложено много алгоритмов перемежения/деперемежения. Наиболее простыми являются блочные. При блочном перемежении входные биты делятся на блоки, которые последовательно записываются в строки некоторой таблицы, приведенной для наглядности на рис. 7.1.



Записываем информационное слово:



Считаем проверочную матрицу, проверочные биты, добавляем их и формируем кодовые слова:



Строим матрицу перемежения, просто построчно записываем. Одно кодовое слово – одна строка.



Дальше кодовая комбинация читается по столбцам и записывается в строку:



Возникает пакет ошибок(черным):



Опять строим матрицу, но это уже деперемежение, записываем кодовые комбинации по столбцам:

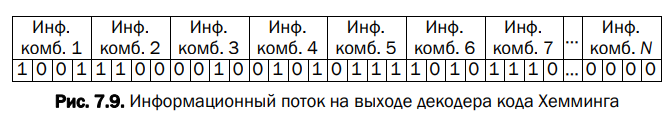


Записываем матрицу деперемежения в одну строку по строкам:



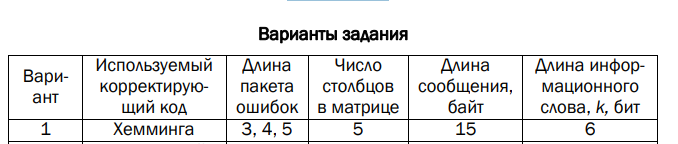
Ошибки исправляем по базе как мы это делали в обычных ситуациях используя матрицу Хемминга.

Исправили ошибки, убрали избыточные биты, получили сообщение:



**Практическое задание**

1. Необходимо разработать авторское приложение в соответствии с целью лабораторной работы. По умолчанию используется блочный перемежитель/деперемежитель. По желанию студент может использовать иной. Задание выполняется по указанию преподавателя в соответствии с вариантом из таблицы.



За основу разрабатываемого приложения может быть взято приложение из выполненной лабораторной работы, соответствующей заданному корректирующему коду.

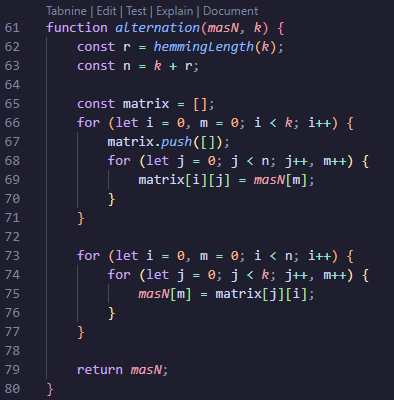


Рисунок 1.1 – Функция для перемежения

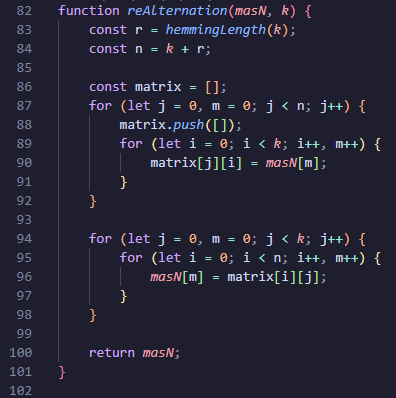


Рисунок 1.2 – Функция для деперемежения

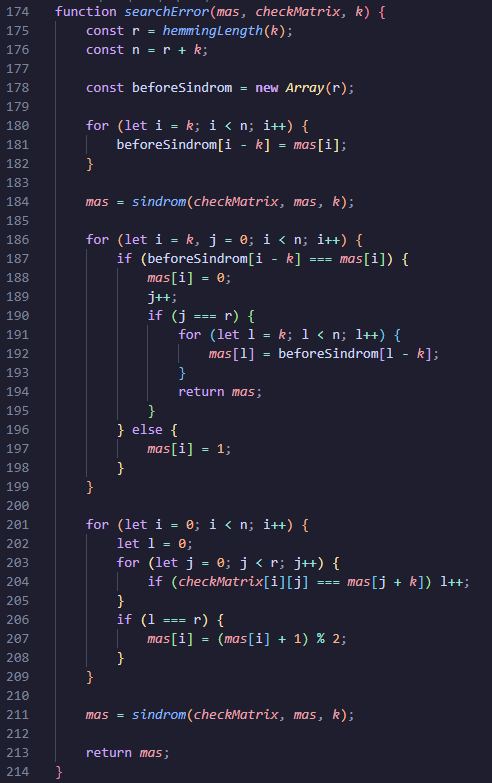
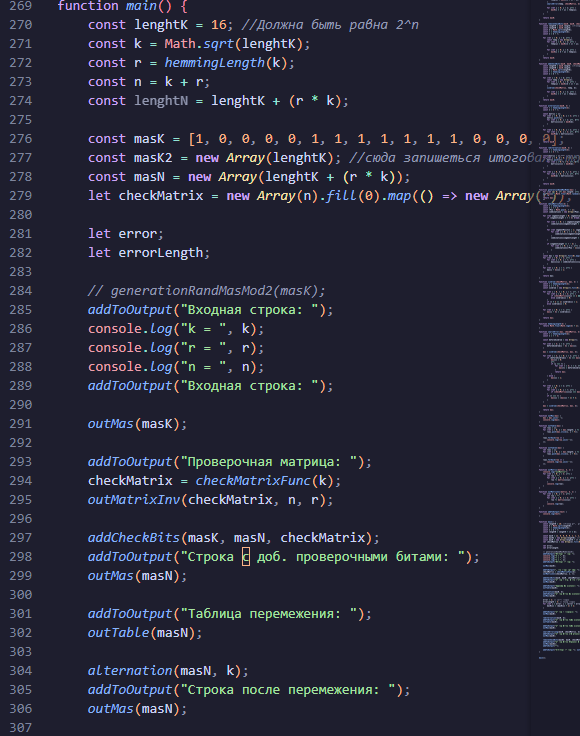


Рисунок 1.3 – Функция поиска ошибок



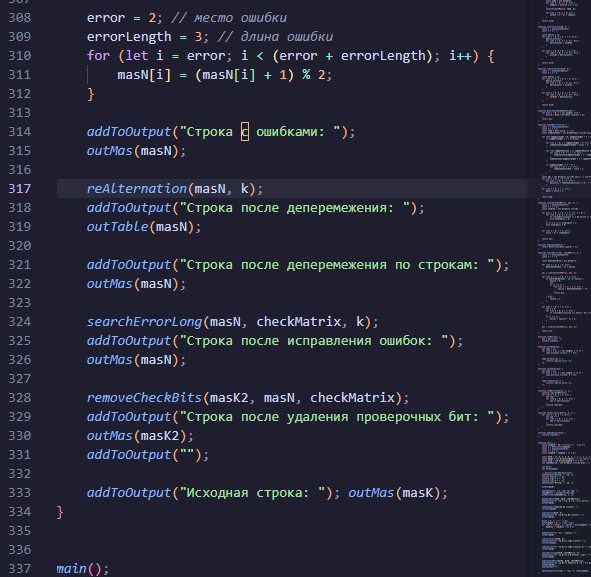


Рисунок 1.4 – Главная функция приложения

2. Местоположение заданной группы ошибок выбирается (генерируется) случайным образом. Необходимо для группы ошибок каждой длины сгенерировать 30−40 случайных ситуаций. После деперемежения и исправления ошибок в сообщении сравнить передаваемую последовательность и полученную после исправления ошибок. Проанализировать эффективность перемежения/деперемежения.

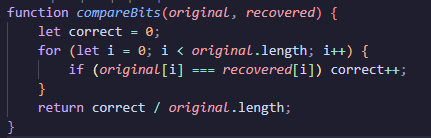


Рисунок 1.5 – Функция подсчета эффективности

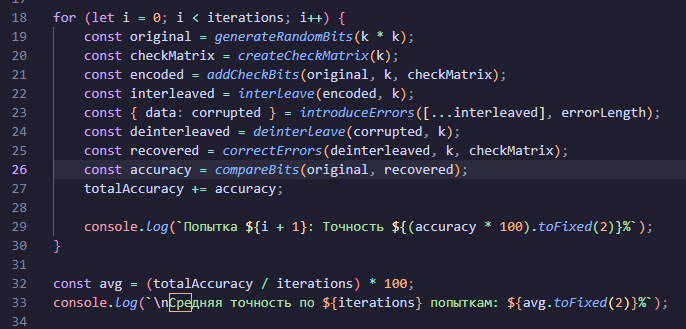


Рисунок 1.6 – Главная функция подсчета эффективности

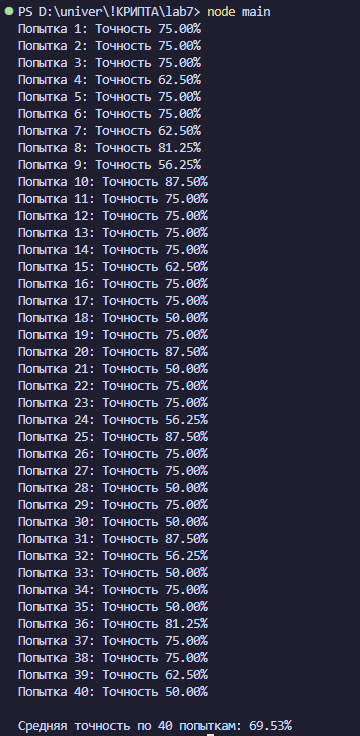


Рисунок 1.7 – Результат работы подсчета эффективности

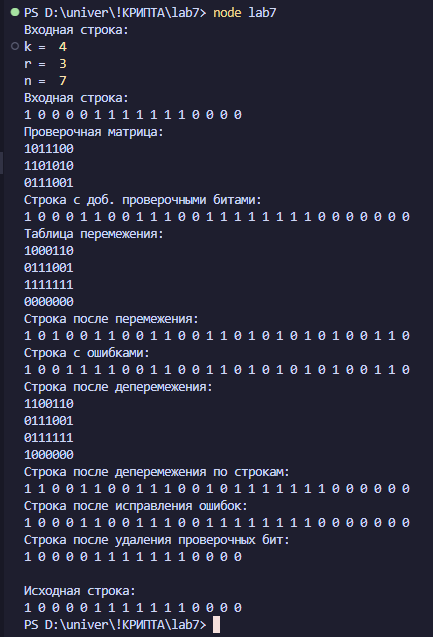


Рисунок 1.8 – Результат работы основного приложения (3 ошибки 3 бит)

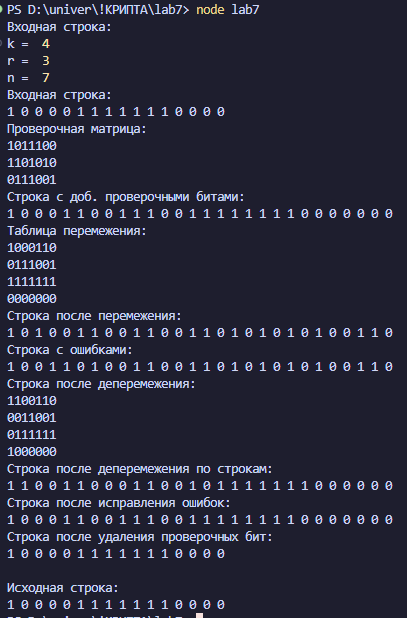


Рисунок 1.9 – Результат работы основного приложения (4 ошибки)

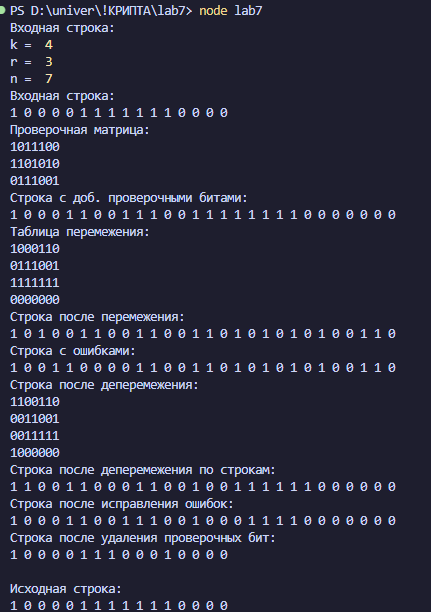


Рисунок 1.10 – Результат работы основного приложения (5 ошибок)

**Вывод**

Точность исправления ошибок: Средняя точность алгоритма составляет 69.53%. Это может свидетельствовать о том, что алгоритм хорошо справляется с исправлением ошибок, однако существуют ситуации, когда исправление не дает идеального результата.

Метод перемежения и деперемежения оказался полезным для повышения точности передачи данных, однако ошибка все же сохраняется, особенно в некоторых случаях.