МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»

(БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа 7

по дисциплине: Теория информации

тема: «RLE»

Выполнил: ст. группы ПВ-211

Шамраев Александр Анатольевич

Проверил:

Твердохлеб Виталий Викторович

Белгород 2023 г.

содержание отчета

[Задание 3](#_Toc134000872)

[1 Выбрать для обработки пару текстовых последовательностей. 4](#_Toc134000873)

[1.1 Hello, World! 4](#_Toc134000874)

[1.2 4](#_Toc134000875)

[2 Построить обработчик структурных особенностей сообщения по методу RLЕ (например, в соответствии с выражением 9 слайдов). 5](#_Toc134000876)

[3 Обработать последовательности структурным методом после чего - с применением метода, указанного в п.3. 6](#_Toc134000877)

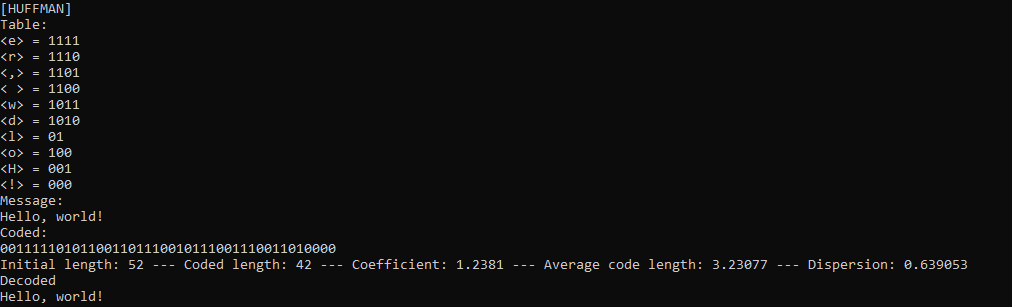
[4 Сравнить полученные показатели коэффициента сжатия и прочие для каждого из исходных сообщений в каждом из режимов обработки. 7](#_Toc134000878)

Задание

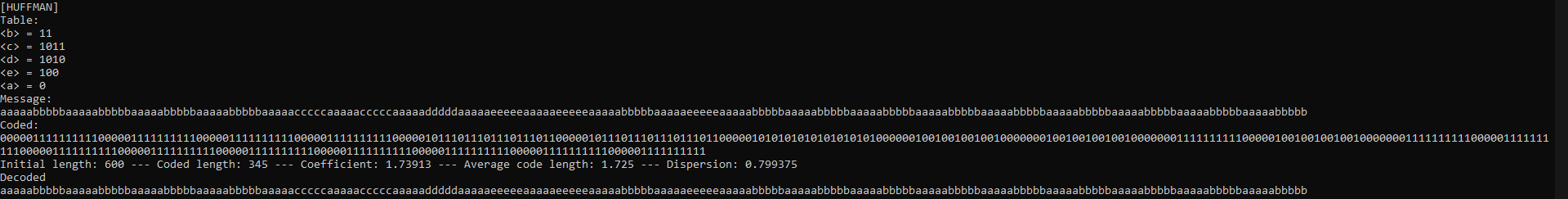
1. Выбрать для обработки пару текстовых последовательностей
2. Построить обработчик структурных особенностей сообщения по методу RLЕ (например, в соответствии с выражением 9 слайдов).
3. Закодировать каждую из последовательностей одним из ранее использованных методов, зафиксировать полученные показатели.
4. Обработать последовательности структурным методом после чего - с применением метода, указанного в п.3.
5. Сравнить полученные показатели коэффициента сжатия и прочие для каждого из исходных сообщений в каждом из режимов обработки.

# Выбрать для обработки пару текстовых последовательностей.

## Hello, World!



## 

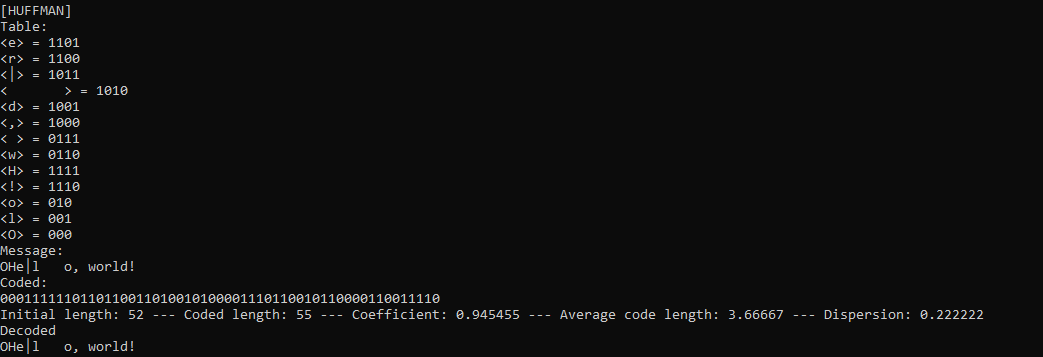


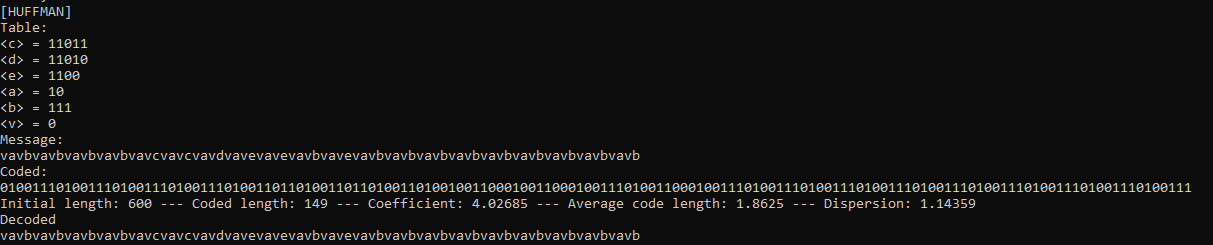
# Построить обработчик структурных особенностей сообщения по методу RLЕ (например, в соответствии с выражением 9 слайдов).

std::string Encode(const std::string &str) {  
 std::string encoded;  
 int count;  
 for (int i = 0; i < str.length();) {  
 if (str[i] != str[i + 1]) {  
 count = 0;  
 for (int j = 0; str[i + j] != str[i + j + 1]; ++j) {  
 count++;  
 }  
 encoded += count;  
 for (int j = 0; j < count; ++j) {  
 encoded += str[i++];  
 }  
 } else {  
 count = -1;  
 while (str[i] == str[i + 1]) {  
 count--, i++;  
 }  
 encoded += count;  
 encoded += str[i++];  
 }  
  
  
 }  
 return encoded;  
}  
  
std::string Decode(const std::string &encoded) {  
 std::string msg;  
 for (int i = 0; i < encoded.length();) {  
 int count = encoded[i];  
 if (count > 0) {  
 for (int j = 0; j < count; ++j) {  
 msg += encoded[i + j + 1];  
 }  
  
 i += count + 1;  
 } else {  
 count = -count;  
 for (int j = 0; j < count; ++j) {  
 msg += encoded[i + 1];  
 }  
  
 i += count;  
 }  
 }  
  
 return msg;  
}

# Обработать последовательности структурным методом после чего - с применением метода, указанного в п.3.







# Сравнить полученные показатели коэффициента сжатия и прочие для каждого из исходных сообщений в каждом из режимов обработки.

Как видно из результатов кодирования – применение совместно методов RLE и Хаффмана эффективно на длинных сообщениях, содержащих длительные последовательности одинаковых символов и различную частоту их появления (к таким видам сообщений можно отнести некоторые виды растровых изображений). Реальный текст при помощи метода RLE не сжимается в принципе (в реальном тексте не существует длительных последовательностей одинаковых символов) и в результате кодирования сообщение растягивается. При некоторых модификациях метода можно свести расширение сообщения к минимуму.