

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»

Выполнил: ст. группы ПВ-211

Чувилко Илья Романович

Проверил:

Твердохлеб В.В.

Белгород 2023 г.

**Тема: «RLE»**

**Содержание отчета**

- 1) Выбрать для обработки пару текстовых последовательностей
- 2) Построить обработчик структурных особенностей сообщения по методу RLE (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.

**Ход работы:**

**Задание 1:** Выбрать для обработки пару текстовых последовательностей.

Hello, World!

```
[HUFFMAN]
Table:
<e> = 1111
<r> = 1110
<, > = 1101
< > = 1100
<w> = 1011
<d> = 1010
<l> = 01
<o> = 100
<H> = 001
<!> = 000
Message:
Hello, world!
Coded:
001111101011001101110010111001110011010000
Initial length: 52 --- Coded length: 42 --- Coefficient: 1.2381 --- Average code length: 3.23077 --- Dispersion: 0.639053
Decoded
Hello, world!
```

$$aaaabbbbbaaaacccc \dots \{160\}$$
[illegible]

**Задание 2:** Построить обработчик структурных особенностей сообщения по методу RLE (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;
}

```

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

```
o, world!  
vavbvavbvavbvavbvavcva/cvavdvavevavevavbvavevavbvavbvavbvavbvavbvavbvavbvavbv
```

```

[HUFFMAN]
Table:
<e> = 1101
<r> = 1100
<|> = 1011
<      > = 1010
<d> = 1001
<,> = 1000
< > = 0111
<w> = 0110
<H> = 1111
<!> = 1110
<o> = 010
<l> = 001
<O> = 000
Message:
OHe|l  o, world!
Coded:
0001111110110110011010010100001110110010110000110011110
Initial length: 52 --- Coded length: 55 --- Coefficient: 0.945455 --- Average code length: 3.66667 --- Dispersion: 0.222222
Decoded
OHe|l  o, world!

```

[illegible]

Задание 4: 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.

