Лабораторная работа №8-9

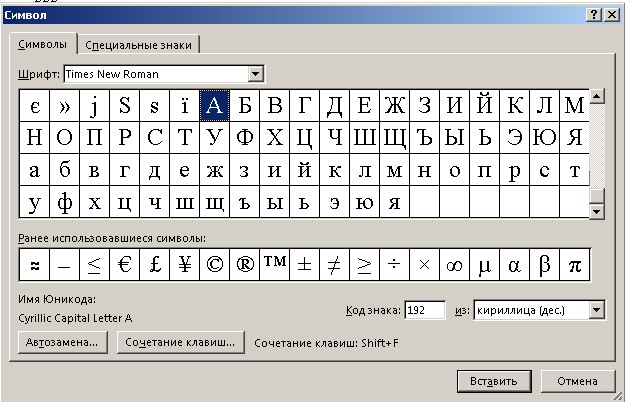
**Сжатие информации методом RLE**

**Задание № 1**

Выполнить вручную кодирование сообщения методом RLE. В качестве исходной фразы взять текст из таблицы. С помощью таблицы ASCII перевести символы заданной фразы в десятичные числа, а затем десятичные числа перевести в двоичные. Выполнить сжатие информации, вычислить контрольные суммы и коэффициент сжатия.

|  |  |
| --- | --- |
| Вариант | Текст |
| 1 | Кредитка 2235555666122 |
| 2 | Паспорт 25700000333215 |
| 3 | ИНН 78888255555488856 |
| 4 | Пароль 177775556666612 |
| 5 | Касса 1478885555233333 |

|  |  |
| --- | --- |
| Вариант | Текст |
| 6 | Пароль abcWWWWZZZq |
| 7 | Автомобиль 78999994441 |
| 8 | Алло это 4565555544488 |
| 9 | Удостоверение 265444111 |
| 10 | Счет 95122244445333333 |



Упакованная методом RLE последовательность состоит из **управляющих байтов**, за которыми следуют один или несколько байтов данных. При этом если старший бит управляющего байта равен 1, то следующий за ним байт данных нужно повторить при декодировании столько раз, сколько указано в оставшихся 7 битах управляющего байта.

Например, управляющий байт 10001001 говорит, что следующий за ним байт нужно повторить 9 раз, так как 10012 = 910.

Если старший бит управляющего байта равен 0, то при декодировании нужно взять несколько следующих байтов без изменений. Число байтов, которые берутся без изменений, указывается в оставшихся 7 битах. Например, управляющий байт 00000011 говорит, что следующие за ним 3 байта нужно взять без изменений.

**Пример сжатия методом RLE.**

Пусть дана последовательность из 12 байтов:

11111111 11111111 11111111 11111111 11111111 11110000

00001111 11000011 10101010 10101010 10101010 10101010.

В начале исходной двоичной последовательности 5 раз повторяется байт 11111111. Чтобы упаковать эти 5 байтов, нужно записать сначала управляющий байт 10000101, а затем повторяемый байт 11111111. В результате сжатия этого фрагмента данных выигрыш составит 3 байта. Далее идут 3 разных (неповторяющихся) байта: 11110000 00001111 и 11000011. Чтобы их «упаковать», нужно записать управляющий байт 00000011, а затем указать эти 3 неповторяющихся байта. В результате архивации этого фрагмента двоичной последовательности получается увеличение объема архива на 1 байт. Далее в последовательности 4 раза повторяется байт 10101010. Для архивации этого фрагмента двоичных данных нужно сформировать управляющий байт 10000100 и записать повторяемый байт 10101010. Сжатие последнего фрагмента даст выигрыш 2 байта.

Число байтов, которые берутся без изменений, указывается в оставшихся 7 битах. Например, управляющий байт 00000011 говорит, что следующие за ним 3 байта нужно взять без изменений.

Рассмотрим пример сжатия методом RLE.

Пусть дана некоторая последовательность из 12 байтов:

11111111 11111111 11111111 11111111 11111111 11110000

00001111 11000011 10101010 10101010 10101010 10101010.

В начале исходной двоичной последовательности 5 раз повторяется байт 11111111. Чтобы упаковать эти 5 байтов, нужно записать сначала управляющий байт 10000101, а затем повторяемый байт 11111111. В результате сжатия этого фрагмента данных выигрыш составит 3 байта. Далее идут 3 разных (неповторяющихся) байта: 11110000 00001111 и 11000011. Чтобы их «упаковать», нужно записать управляющий байт 00000011, а затем указать эти 3 неповторяющихся байта. В результате архивации этого фрагмента двоичной последовательности получается увеличение объема архива на 1 байт. Далее в последовательности 4 раза повторяется байт 10101010. Для архивации этого фрагмента двоичных данных нужно сформировать управляющий байт 10000100 и записать повторяемый байт 10101010. Сжатие последнего фрагмента даст выигрыш 2 байта.

В результате такой архивации получена новая последовательность данных (архив), состоящая из 8 байтов:

10000101 11111111 00000011 11110000

00001111 11000011 10000100 10101010.

Таким образом, 12 байт исходной двоичной последовательности удалось сжать до 8 байт. Коэффициент сжатия равен 12/8=1,5.

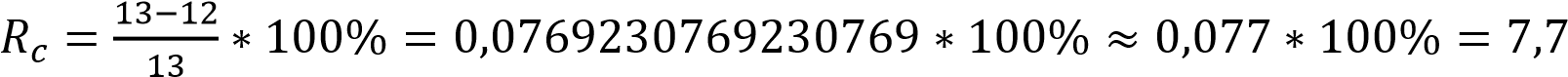
Выполним сжатие сообщения методом RLE. Текст сообщения: **ИНН 222221333.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Фраза | Десятичный код | **№** | **Двоичны код** | | **й** | **Архив** | |  | **№** |
| И | 200 | 1 |  | 11001000 |  | **00000001** | |  | 1 |
| Н | 205 | 2 |  | 11001101 |  |  | 11001000 |  | 2 |
| Н | 205 | 3 |  | 11001101 |  | **10000010** | |  | 3 |
|  | 32 | 4 |  | 00100000 |  |  | 11001101 |  | 4 |
| 2 | 50 | 5 |  | 00110010 |  | **00000001** | |  | 5 |
| 2 | 50 | 6 |  | 00110010 |  |  | 00100000 |  | 6 |
| 2 | 50 | 7 |  | 00110010 |  | **10000101** | |  | 7 |
| 2 | 50 | 8 |  | 00110010 |  |  | 00110010 |  | 8 |
| 2 | 50 | 9 |  | 00110010 |  | **00000001** | |  | 9 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 49 | 10 |  | 00110001 |  |  | 00110001 |  | 10 |
| 3 | 51 | 11 |  | 00110011 |  | **10000011** | |  | 11 |
| 3 | 51 | 12 |  | 00110011 |  |  | 00110011 |  | **12** |
| 3 | 51 | **13** |  | 00110011 |  |  | |  |  |
| КС двоичная |  |  |  | |  | 10010000B | |  |  |
| КС шестнадцатеричная |  |  |  | |  | 90H | |  |  |

Для упрощения проверки результата сжатия были вычислены контрольные суммы (КС) в двоичной и шестнадцатеричной системах счисления.

Коэффициент сжатия в данном случае составил: 13/12=1,08.

Таким образом, тринадцать байт исходного текста сжаты до двенадцать байт. Степень сжатия R, характеризующая относительное уменьшение объема данных:

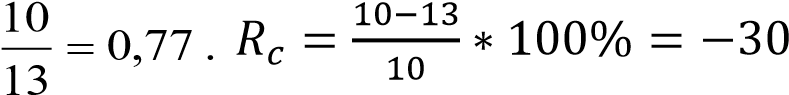
Ответ: *Кс =*1,08, *R*c=7,7

**Задание № 2**

Распаковать информацию по методу RLE и расшифровать слово

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | **00000010** |  |  |  |  |
| 2 | 11000001 |  |  |  |  |
| 3 | 11000101 |  |  |  |  |
| 4 | **10000010** |  |  |  |  |
| 5 | 11001011 |  |  |  |  |
| 6 | **00000011** |  |  |  |  |
| 7 | 11000000 |  |  |  |  |
| 8 | 11000100 |  |  |  |  |
| 9 | 11001110 |  |  |  |  |
| 10 | **10000010** |  |  |  |  |
| 11 | 11001101 |  |  |  |  |
| 12 | **00000001** |  |  |  |  |
| 13 | 11000000 |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | **00000010** | 11000001 | 193 | Б | 1 |
| 2 | 11000001 | 11000101 | 197 | Е | 2 |
| 3 | 11000101 | 11001011 | 203 | Л | 3 |
| 4 | **10000010** | 11001011 | 203 | Л | 4 |
| 5 | 11001011 | 11000000 | 192 | А | 5 |
| 6 | **00000011** | 11000100 | 196 | Д | 6 |
| 7 | 11000000 | 11001110 | 206 | О | 7 |
| 8 | 11000100 | 11001101 | 205 | Н | 8 |
| 9 | 11001110 | 11001101 | 205 | Н | 9 |
| 10 | **10000010** | 11000000 | 192 | А | 10 |
| 11 | 11001101 |  |  |  |  |
| 12 | **00000001** |  |  |  |  |
| 13 | 11000000 |  |  |  |  |

Коэффициент сжатия =

Ответ: БЕЛЛАДОННА. Вывод: Для текстовых данных методы *RLE,* как правило, не эффективны.

**Варианты заданий**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант №1, 6 | Вариант № 2, 7 | Вариант № 3, 8 | Вариант № 4, 9 | Вариант № 5, 10 |
| |  |  | | --- | --- | | 1 | **00000010** | | 2 | 11001010 | | 3 | 11001110 | | 4 | **10000010** | | 5 | 11001010 | | 6 | **00000101** | | 7 | 11001110 | | 8 | 11000001 | | 9 | 11000000 | | 10 | 11010110 | | 11 | 11001000 | | 12 | **10000010** | | 13 | 11001011 | | 14 | **00000001** | | 15 | 11000000 | | |  |  | | --- | --- | | 1 | **00000010** | | 2 | 11000001 | | 3 | 11000101 | | 4 | **10000010** | | 5 | 11010001 | | 6 | **00000001** | | 7 | 11001110 | | 8 | **10000010** | | 9 | 11001101 | | 10 | **00000011** | | 11 | 11001000 | | 12 | 11010110 | | 13 | 11000000 | | |  |  | | --- | --- | | 1 | **00000011** | | 2 | 11001110 | | 3 | 11000001 | | 4 | 11001110 | | 5 | **10000010** | | 6 | 11000110 | | 7 | **00000001** | | 8 | 11000101 | | 9 | **10000010** | | 10 | 11001101 | | 11 | **00000010** | | 12 | 11011011 | | 13 | 11001001 | | |  |  | | --- | --- | | 1 | **00000010** | | 2 | 11000010 | | 4 | 11011011 | | 5 | **10000010** | | 6 | 11000110 | | 7 | **00000001** | | 8 | 11000101 | | 9 | **10000010** | | 10 | 11001101 | | 11 | **00000010** | | 12 | 11011011 | | 13 | 11001001 | | |  |  | | --- | --- | | 1 | **00000010** | | 2 | 11000111 | | 4 | 11000000 | | 5 | **10000010** | | 6 | 11000110 | | 7 | **00000001** | | 8 | 11000101 | | 9 | **10000010** | | 10 | 11001101 | | 11 | **00000010** | | 12 | 11011011 | | 13 | 11001001 | |