## Практическая работа

## Алгоритм RLE

1. Используя алгоритм RLE, закодируйте последовательность символов

**BBBBBBACCCABBBBBB**

Запишите результат в виде шестнадцатеричных кодов (каждый символ кодируется в виде байта, который представлен двумя шестнадцатеричными цифрами ). Проверьте полученный результат с помощью программы RLE.

*Ответ*:

1. Раскодируйте последовательность, упакованную с помощью алгоритма RLE (приводятся шестнадцатеричные коды): 01 4D 8E 41 01 4D 8E 4116. Для определения символов по их шестнадцатеричным кодом используйте таблицу ASCII. В приведённой таблице в первом столбце записана первая цифра шестнадцатеричного кода символа, а в первой строке – вторая. Например, символ «&» имеет шестнадцатеричный код 2616.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **.0** | **.1** | **.2** | **.3** | **.4** | **.5** | **.6** | **.7** | **.8** | **.9** | **.A** | **.B** | **.C** | **.D** | **.E** | **.F** |
| **0.** | NUL | SOH | STX | ETX | EOT | ENQ | ACK | BEL | BS | TAB | LF | VT | FF | CR | SO | SI |
| **1.** | DLE | DC1 | DC2 | DC3 | DC4 | NAK | SYN | ETB | CAN | EM | SUB | ESC | FS | GS | RS | US |
| **2.** |  | ! | " | # | $ | % | & | ' | ( | ) | \* | + | , | — | . | / |
| **3.** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| **4.** | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| **5.** | P | Q | R | S | T | U | V | W | X | Y | Z | [ | \ | ] | ^ | \_ |
| **6.** | ` | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| **7.** | p | q | r | s | t | u | v | w | x | y | z | { | | | } | ~ | DEL |

*Ответ*:

1. Определите количество байтов в исходной и распакованной последовательности (см. предыдущее задание) и вычислите коэффициент сжатия:

|  |  |  |
| --- | --- | --- |
| **Сжатая последовательность** | **Несжатая последовательность** | **Коэффициент сжатия** |
|  |  |  |

1. Проверьте результат, полученный в предыдущем пункте, с помощью программы RLE. Предложите два способа проверки.
2. Постройте последовательности, которые сжимаются алгоритмом RLE ровно в 2 раза, в 4 раза, в 5 раз. Проверьте свои ответы с помощью программы RLE.

|  |  |  |
| --- | --- | --- |
| **Несжатая последовательность** | **Сжатая последовательность** | **Коэффициент сжатия** |
| AAAA |  |  |
| AAAAAAAA |  |  |
| AAAAAAAAAA |  |  |

1. Придумайте три последовательности, которые невозможно сжать с помощью алгоритма RLE:

|  |  |  |
| --- | --- | --- |
| **Несжатая последовательность** | **«Сжатая» последовательность** | **Коэффициент сжатия** |
| M |  |  |
| MA |  |  |
| POP |  |  |

1. Используя программу RLE, примените RLE-сжатие к следующим файлам и найдите для каждого из них коэффициент сжатия:

|  |  |  |  |
| --- | --- | --- | --- |
| **Файл** | **Размер без сжатия** | **Размер после сжатия** | **Коэффициент сжатия** |
| **grad\_vert.bmp** |  |  |  |
| **grad\_horz.bmp** |  |  |  |
| **grad\_diag.jpg** |  |  |  |

1. Объясните результаты, полученные в предыдущем пункте:
   * почему не удается сжать рисунки в формате JPEG?

*Ответ*:

* + почему для двух рисунков в формате BMP одинакового размера коэффициенты сжатия по алгоритму RLE так сильно отличаются? *Подсказка*: откройте эти рисунки в любой программе просмотра.

*Ответ*:

1. Оцените максимально достижимый коэффициент сжатия с помощью рассмотренного в учебнике варианта RLE-алгоритма. В каком случае его удастся достичь?

*Ответ*:

1. Оцените коэффициент сжатия с помощью RLE-алгоритма в худшем случае. Опишите этот худший случай.

*Ответ*: