```
// Написано в FreeBSD, Midnight Commander.
// Ukraine. (C) Demidov S.V.

// JavaScript.

// ---------------------- //
// Кодирование VRLE8 (NEW) //
// ---------------------- //

//
// Coding.
// Объединённые две функции (две функции в одну).
//
function VRLE8NewCoding(buffer)
{

//
// Входной массив buffer (каждая ячейка от 0 и до 255).
// Выходной массив $arraycoding8 (каждая ячейка от 0 и до 255).
// Работаем с байтами!
//

//
// Как могут располагаться повтор.
//
//    0, 0, 1, 2, 3, 4 => В начале.
//    1, 2, 3, 4, 0, 0 => В конце.
//    1, 2, 0, 0, 1, 2 => Внутри.
//    0, 0, 0, 0, 0, 0 => Только одни повтор.
//    0, 0, 1, 1, 2, 3 => Стоящие рядом.
//
// Как могут располагаться неповтор.
//
//    1, 2, 0, 0, 0, 0 => В начале.
//    0, 0, 0, 0, 1, 2 => В конце.
//    0, 0, 1, 2, 0, 0 => Внутри.
//    1, 2, 3, 4, 5, 6 => Только одни неповтор.
//
//    С одним байтом.
//
//    1, 0, 0, 0, 0, 0 => В начале.
//    0, 0, 0, 0, 0, 1 => В конце.
//    0, 0, 1, 0, 0, 0 => Внутри.
//

// Этот код можно оптимизировать!

   var bl, addr2, addr3, total, p1;

// addr2 - адрес первого повтор./неповтор. в buffer-массиве.
// addr3 - адрес последнего повтор./неповтор. в buffer-массиве.
// total - сколько всех повтор./неповтор.
//    bl - здесь размер buffer-массива.

//                                        ------ Повтор. -------
//                                        |                    |
//                  ------------------------------------------------------------------
// buffer (массив): | Байт | Байт | Байт | 0x00 | 0x00 | 0x00 | 0x00 | Байт | Байт | Байт | ...
//                  ------------------------------------------------------------------
//                                        |                    |
//                                      addr2                addr3
//
//                                        ----- Неповтор. ------
//                                        |                    |
//                  ------------------------------------------------------------------
// buffer (массив): | Байт | Байт | Байт | 0x01 | 0x02 | 0x03 | 0x04 | Байт | Байт | Байт | ...
//                  ------------------------------------------------------------------
//                                        |                    |
//                                      addr2                addr3
//
// Для одного байта (неповтор.): addr2 = addr3.

   // Переменные кодера.
   var $addrcoding, $z1, $z2, $z3, $b1;
   var $tpor1, $tpor2, $swpor, $t1, $t2, $z4;

   // Кодируем в arraycoding8.
   var $arraycoding8 = [];

   $addrcoding = 0;
```

```
// Размер buffer'a.
bl = buffer.length;

if (bl == 1)
   {
   // buffer содержит всего 1 байт.

   $arraycoding8[0] = 1;
   $arraycoding8[1] = buffer[0];

   }
   else
   {
   for (addr3 = 0;;) // Бесконечный цикл.
      {
      // Достигнут конец buffer'a.
      if (addr3 == bl)
         {
         // Прервать цикл.
         break;
         }

      // В конце buffer'a один байт.
      if (addr3 == bl - 1)
         {
         // Переменные на выходе:
         // В addr2 адрес первого неповтор.
         // В addr3 адрес последнего неповтор.
         // В total сколько всех неповтор.

         // addr2 = addr3; total = 1;

         $arraycoding8[$addrcoding++] = 1;
         $arraycoding8[$addrcoding] = buffer[addr3];

         // Прервать цикл.
         break;
         }

      // (!)
      if (buffer[addr3] == buffer[addr3 + 1])
         {
         // Повторяющиеся.

         // В addr2 адрес первого повтор.
         addr2 = addr3;

         // Запомнить в p1.
         p1 = buffer[addr3];

         for (total = 1;;) // Бесконечный цикл.
            {
            // Достигнут конец buffer'a.
            if (addr3 == bl - 1)
               {
               // Прервать цикл.
               break;
               }

            // Если идут повтор., то считаем их.
            if (p1 == buffer[addr3 + 1])
               {
               addr3++;

               // В total сколько повтор.
               total++;
               }
               else
               {
               // Прервать цикл.
               break;
               }
            }

         // Переменные на выходе:
         // В addr2 адрес первого повтор.
         // В addr3 адрес последнего повтор.
         // В total сколько всех повтор.

         // --- Кодирование повтор. ---
         // -------------------------
```

```
// Минимальная порция - 2 байта.
// Максимальная порция - 127 байт.

// Байт, который нужно повторить.
// $b1 = buffer[addr2];

// Сколько всех повтор.
$tpor1 = total; // Это можно оптимизировать.

if ($tpor1 < 127)
   {
   // Только часть порции.
   // В $tpor1 - сколько повтор.
   $swpor = 1;
   }
   else
   {
   $t1 = 0;

   for (;;)
      {
      $tpor1 = $tpor1 - 127;
      $t1++;

      if ($tpor1 == 0)
         {
         // n-полных порций и нет неполной порции.
         // В $t1 - сколько полных порций.
         $swpor = 2;
         break;
         }

      if ($tpor1 < 127)
         {
         // n-полных порций и часть порции.
         // В $t1 - сколько полных порций.
         // В $tpor1 - часть порции (сколько повтор.).
         $swpor = 3;
         break;
         }
      }
   }

switch ($swpor)
   {
   case 1:

   // Логическое ИЛИ (|):
   //    0 0 | 0
   //    0 1 | 1
   //    1 0 | 1
   //    1 1 | 1

   // Только часть порции.

   // Записываем информационный байт.
   // Установить 7-й бит в 1, остальные биты оставить без изменений.
   $arraycoding8[$addrcoding++] = 128 | $tpor1;

   // Записываем байт, который нужно повторить.
   $arraycoding8[$addrcoding++] = $b1;

   break;

   case 2:

   // Только полная порция (полные порции).
   for ($z1 = 0; $z1 < $t1;)
      {
      // Записываем информационный байт.
      // 7-й бит в 1 (7F => FF)!
      $arraycoding8[$addrcoding++] = 255;

      // Записываем байт, который нужно повторить.
      $arraycoding8[$addrcoding++] = $b1;

      $z1++;
      }

   break;
```

```
            case 3:

            // Полная порция (полные порции).
            for ($z1 = 0; $z1 < $t1;)
               {
               // Записываем информационный байт.
               // 7-й бит в 1 (7F => FF)!
               $arraycoding8[$addrcoding++] = 255;

               // Записываем байт, который нужно повторить.
               $arraycoding8[$addrcoding++] = $b1;

               $z1++;
               }

            // Часть порции (остаток).

            // Записываем информационный байт.
            // Установить 7-й бит в 1, остальные биты оставить без изменений.
            $arraycoding8[$addrcoding++] = 128 | $tpor1;

            // Записываем байт, который нужно повторить.
            $arraycoding8[$addrcoding++] = $b1;

            break;
            } // Конец switch $swpor.

         }
         else
         {
         // Неповторяющиеся.

         // В addr2 адрес первого неповтор.
         addr2 = addr3;

         for (total = 1;;) // Бесконечный цикл.
            {
            // Достигнут конец buffer'a.
            if (addr3 == bl - 1)
               {
               // Прервать цикл.
               break;
               }

            // Продолжаем искать повтор.
            if (buffer[addr3] == buffer[addr3 + 1])
               {
               // Откатится назад (к концу неповтор.).
               total--;
               addr3--;

               // Прервать цикл.
               break;
               }
               else
               {
               addr3++;

               // В total сколько неповтор.
               total++;
               }
            }

         // Переменные на выходе:
         // В addr2 адрес первого неповтор.
         // В addr3 адрес последнего неповтор.
         // В total сколько всех неповтор.

         // --- Кодирование неповтор. ---
         // ----------------------------

         // Минимальная порция - 1 байт.
         // Максимальная порция - 127 байт.

         // Начальный адрес неповтор.
         $z3 = addr2;

         // Сколько неповтор.
         $tpor2 = total; // Это можно оптимизировать.

         if ($tpor2 < 127)
            {
```

```
            // Только часть порции.
            // В $tpor2 - сколько неповтор.
            $swpor = 1;
            }
            else
            {
            $t2 = 0;

            for (;;)
               {
               $tpor2 = $tpor2 - 127;
               $t2++;

               if ($tpor2 == 0)
                  {
                  // n-полных порций и нет неполной порции.
                  // В $t2 - сколько полных порций.
                  $swpor = 2;
                  break;
                  }

               if ($tpor2 < 127)
                  {
                  // n-полных порций и часть порции.
                  // В $t2 - сколько полных порций.
                  // В $tpor2 - часть порции (сколько неповтор.).
                  $swpor = 3;
                  break;
                  }
               }
            }

         switch ($swpor)
            {
            case 1:

            // Записываем информационный байт.
            // 7-й бит уже в 0!
            $arraycoding8[$addrcoding++] = $tpor2;

            // Только часть порции.
            for ($z2 = 0; $z2 < $tpor2;)
               {
               // Переписываем неповтор. байты.
               $arraycoding8[$addrcoding++] = buffer[$z3++];
               $z2++;
               }

            break;

            case 2:

            // Только полная порция (только полные порции).
            for ($z4 = 0; $z4 < $t2;)
               {
               // Записываем информационный байт.
               // Максимальный размер порции, плюс 7-й бит в 0.
               $arraycoding8[$addrcoding++] = 127;

               for ($z2 = 0; $z2 < 127;)
                  {
                  // Переписываем неповтор. байты.
                  $arraycoding8[$addrcoding++] = buffer[$z3++];
                  $z2++;
                  }
               $z4++;
               }

            break;

            case 3:

            // Полная порция (полные порции).
            for ($z4 = 0; $z4 < $t2;)
               {
               // Записываем информационный байт.
               // Максимальный размер порции, плюс 7-й бит в 0.
               $arraycoding8[$addrcoding++] = 127;

               for ($z2 = 0; $z2 < 127;)
                  {
                  // Переписываем неповтор. байты.
```

```
                    $arraycoding8[$addrcoding++] = buffer[$z3++];
                    $z2++;
                    }
                 $z4++;
                 }

              // Часть порции (остаток).

              // Записываем информационный байт.
              // 7-й бит уже в 0!
              $arraycoding8[$addrcoding++] = $tpor2;

              for ($z2 = 0; $z2 < $tpor2;)
                 {
                 // Переписываем неповтор. байты.
                 $arraycoding8[$addrcoding++] = buffer[$z3++];
                 $z2++;
                 }

              break;
              } // Конец switch $swpor.

           } // Конец неповтор.

        addr3++;

        } // Конец for (бесконечный цикл).

     } // Конец if.

  // Вернуть закодированный массив.
  return $arraycoding8;
}
```