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

// JavaScript.

// --------------------- //
// Кодування VRLE8 (NEW) //
// --------------------- //

//
// Coding.
// Об'єднані дві функції (дві функції в одну).
//
function VRLE8NewCoding(buffer)
{

//
// Вхідний масив buffer (кожна комірка від 0 і до 255).
// Вихідний масив $arraycoding8 (кожна комірка від 0 і до 255).
// Працюємо з байтами!
//

//
// Як можуть бути повтор.
//
//    0, 0, 1, 2, 3, 4 => На початку.
//    1, 2, 3, 4, 0, 0 => У кінці.
//    1, 2, 0, 0, 1, 2 => Всередині.
//    0, 0, 0, 0, 0, 0 => Лише одні повтор.
//    0, 0, 1, 1, 2, 3 => Ті, що стоять поруч.
//
// Як можуть розташовуватись неповтор.
//
//    1, 2, 0, 0, 0, 0 => На початку.
//    0, 0, 0, 0, 1, 2 => У кінці.
//    0, 0, 1, 2, 0, 0 => Всередині.
//    1, 2, 3, 4, 5, 6 => Лише одні неповтор.
//
//    З одним байтом.
//
//    1, 0, 0, 0, 0, 0 => На початку.
//    0, 0, 0, 0, 0, 1 => У кінці.
//    0, 0, 1, 0, 0, 0 => Всередині.
//

// Цей код можна оптимізувати!

   var bl, addr2, addr3, total, p1;

// addr2 - адреса першого повтор./неповтор. у buffer-масиві.
// addr3 - адреса останнього повтор./неповтор. у buffer-масиві.
// total - скільки всіх повтор./неповтор.
//    bl - тут розмір buffer-масиву.

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

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

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

   $addrcoding = 0;
```

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

if (bl == 1)
   {
   // buffer містить лише 1 байт.

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

   }
   else
   {
   for (addr3 = 0;;) // Безкінечний цикл.
      {
      // Досягнуто кінця buffer'a.
      if (addr3 == bl)
         {
         // Перервати цикл.
         break;
         }

      // Наприкінці buffer'a один байт.
      if (addr3 == bl - 1)
         {
         // Змінні на виході:
         // В addr2 адреса першого неповтор.
         // В addr3 адреса останнього неповтор.
         // У total скільки всіх неповтор.

         // addr2 = addr3; total = 1;

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

         // Перервати цикл.
         break;
         }

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

         // В addr2 адреса першого повтор.
         addr2 = addr3;

         // Запам'ятати у p1.
         p1 = buffer[addr3];

         for (total = 1;;) // Безкінечний цикл.
            {
            // Досягнуто кінець buffer'a.
            if (addr3 == bl - 1)
               {
               // Перервати цикл.
               break;
               }

            // Якщо йдуть повтор., то рахуємо їх.
            if (p1 == buffer[addr3 + 1])
               {
               addr3++;

               // У total скільки повтор.
               total++;
               }
               else
               {
               // Перервати цикл.
               break;
               }
            }

         // Змінні на виході:
         // В addr2 адреса першого повтор.
         // В addr3 адреса останнього повтор.
         // У total скільки всіх повтор.

         // --- Кодування повтор. ---
         // -------------------------
```

```
// Мінімальна порція – 2 байти.
// Максимальна порція – 127 байт.

// Байт, який слід повторити.
$b1 = buffer[addr2];

// Скільки всіх повтор.
$tpor1 = total; // Це можна оптимізувати.

if ($tpor1 < 127)
   {
   // Лише частина порції.
   // У $tpor1 – скільки повтор.
   $swpor = 1;
   }
   else
   {
   $t1 = 0;

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

      if ($tpor1 == 0)
         {
         // n-повних порцій та немає неповної порції.
         // У $t1 – скільки повних порцій.
         $swpor = 2;
         break;
         }

      if ($tpor1 < 127)
         {
         // n-повних порцій та частина порції.
         // У $t1 – скільки повних порцій.
         // У $tpor1 – частина порції (скільки повтор.).
         $swpor = 3;
         break;
         }
      }
   }

switch ($swpor)
   {
   case 1:

   // Логічне АБО (|):
   //    0 0 | 0
   //    0 1 | 1
   //    1 0 | 1
   //    1 1 | 1

   // Лише частина порції.

   // Записуємо інформаційний байт.
   // Встановити 7 біт в 1, інші біти залишити без змін.
   $arraycoding8[$addrcoding++] = 128 | $tpor1;

   // Записуємо байт, який потрібно повторити.
   $arraycoding8[$addrcoding++] = $b1;

   break;

   case 2:

   // Лише повна порція (повні порції).
   for ($z1 = 0; $z1 < $t1;)
      {
      // Записуємо інформаційний байт.
      // 7-й біт в 1 (7F => FF)!
      $arraycoding8[$addrcoding++] = 255;

      // Записуємо байт, який потрібно повторити.
      $arraycoding8[$addrcoding++] = $b1;

      $z1++;
      }

   break;
```

```
            case 3:

            // Повна порція (повні порції).
            for ($z1 = 0; $z1 < $t1;)
               {
               // Записуємо інформаційний байт.
               // 7-й біт в 1 (7F => FF)!
               $arraycoding8[$addrcoding++] = 255;

               // Записуємо байт, який потрібно повторити.
               $arraycoding8[$addrcoding++] = $b1;

               $z1++;
               }

            // Частина порції (залишок).

            // Записуємо інформаційний байт.
            // Встановити 7 біт в 1, інші біти залишити без змін.
            $arraycoding8[$addrcoding++] = 128 | $tpor1;

            // Записуємо байт, який потрібно повторити.
            $arraycoding8[$addrcoding++] = $b1;

            break;
            } // Кінець switch $swpor.

         }
         else
         {
         // Неповторні.

         // В addr2 адреса першого неповтор.
         addr2 = addr3;

         for (total = 1;;) // Безкінечний цикл.
            {
            // Досягнуто кінця buffer'a.
            if (addr3 == bl - 1)
               {
               // Перервати цикл.
               break;
               }

            // Продовжуємо шукати повтор.
            if (buffer[addr3] == buffer[addr3 + 1])
               {
               // Відкотиться назад (до кінця неповтор.).
               total--;
               addr3--;

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

               // У total скільки неповтор.
               total++;
               }
            }

         // Змінні на виході:
         // В addr2 адреса першого неповтор.
         // В addr3 адреса останнього неповтор.
         // У total скільки всіх неповтор.

         // --- Кодування неповтор. ---
         // --------------------------

         // Мінімальна порція — 1 байт.
         // Максимальна порція — 127 байт.

         // Початкова адреса неповтор.
         $z3 = addr2;

         // Скільки неповтор.
         $tpor2 = total; // Це можна оптимізувати.
```

```
if ($tpor2 < 127)
   {
   // Лише частина порції.
   // У $tpor2 – скільки неповтор.
   $swpor = 1;
   }
   else
   {
   $t2 = 0;

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

      if ($tpor2 == 0)
         {
         // n-повних порцій та немає неповної порції.
         // У $t2 – скільки повних порцій.
         $swpor = 2;
         break;
         }

      if ($tpor2 < 127)
         {
         // n-повних порцій та частина порції.
         // У $t2 – скільки повних порцій.
         // У tpor2 - частина порції (скільки неповтор.).
         $swpor = 3;
         break;
         }
      }
   }

switch ($swpor)
   {
   case 1:

   // Записуємо інформаційний байт.
   // 7-й біт вже 0!
   $arraycoding8[$addrcoding++] = $tpor2;

   // Лише частина порції.
   for ($z2 = 0; $z2 < $tpor2;)
      {
      // Переписуємо неповтор. байти.
      $arraycoding8[$addrcoding++] = buffer[$z3++];
      $z2++;
      }

   break;

   case 2:

   // Лише повна порція (тільки повні порції).
   for ($z4 = 0; $z4 < $t2;)
      {
      // Записуємо інформаційний байт.
      // Максимальний розмір порції, плюс 7-й біт в 0.
      $arraycoding8[$addrcoding++] = 127;

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

   break;

   case 3:

   // Повна порція (повні порції).
   for ($z4 = 0; $z4 < $t2;)
      {
      // Записуємо інформаційний байт.
      // Максимальний розмір порції, плюс 7-й біт в 0.
      $arraycoding8[$addrcoding++] = 127;

      for ($z2 = 0; $z2 < 127;)
```

```
                    {
                    // Переписуємо неповтор. байти.
                    $arraycoding8[$addrcoding++] = buffer[$z3++];
                    $z2++;
                    }
                 $z4++;
                 }

              // Частина порції (залишок).

              // Записуємо інформаційний байт.
              // 7-й біт вже 0!
              $arraycoding8[$addrcoding++] = $tpor2;

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

              break;
              } // Кінець switch $swpor.

           } // Кінець неповтор.

        addr3++;

        } // Кінець for (безкінечний цикл).

     } // Кінець if.

  // Повернути закодований масив.
  return $arraycoding8;
}
```