

```

//
// Построить массив с адресами (повтор. и неповтор.).
//
// Возвращает массив: | x | addr1 | addr2 | y | ... .. и т.д.
//
//
// x = 0 - означает неповтор;
// x = 1 - означает повтор;
// addr1 - начальный адрес повтор./неповтор;
// addr2 - конечный адрес повтор./неповтор;
// y - сколько повтор./неповтор.
//
function VRLE8BuildArrayAddr8(buffer)
{
// buffer - это массив с числами (каждая ячейка от 0 и до 255, байты)!
// buffer - это тот массив, который нужно будет "подсунуть" кодировщику!

//
// 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.
//

var array1 = []; // Выходной массив с адресами повтор. и неповтор.
var x, z, z1, zz, bl, aaddr1;

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

aaddr1 = 0;

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

z = 0; zz = z; x = 1;

// Для проверки.
// alert("Адрес первого неповтор. " + zz + ". " + "Адрес последнего неповтор. " + z + ". " +
// "Сколько неповтор. " + x + ".");

// В массив указатель на неповтор.
array1[aaddr1++] = 0;
// В массив адрес первого неповтор.
array1[aaddr1++] = zz;
// В массив адрес последнего неповтор.
array1[aaddr1++] = z;
// В массив сколько неповтор.
array1[aaddr1] = x;
}
else
{
for (z = 0;;) // Бесконечный цикл.
{
// Достигнут конец buffer'a.
if (z == bl)
{
// Прервать цикл.
break;
}

// В конце buffer'a один байт.

```

```

if (z == bl - 1)
{
    zz = z; x = 1;

    // Для проверки.
    alert("Адрес первого неповтор. " + zz + ". " + "Адрес последнего неповтор. " + z +
    // ". " + "Сколько неповтор. " + x + ".");

    // В массив указатель на неповтор.
    array1[aaddr1++] = 0;
    // В массив адрес первого неповтор.
    array1[aaddr1++] = zz;
    // В массив адрес последнего неповтор.
    array1[aaddr1++] = z;
    // В массив сколько неповтор.
    array1[aaddr1] = x;

    // Прервать цикл.
    break;
}

// Один раз можно выполнить if, а потом сделать через switch.
// Надо подумать!
if (buffer[z] == buffer[z + 1])
{
    // Повторяющиеся.

    // В zz адрес первого повтор.
    zz = z;

    // Запомнить в z1.
    z1 = buffer[z];

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

        // Если идут повтор., то считаем их.
        if (z1 == buffer[z + 1])
        {
            z++;

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

    // Для проверки.
    alert("Адрес первого повтор. " + zz + ". " + "Адрес последнего повтор. " + z + ".
    // " + "Сколько повтор. " + x + ".");

    // В массив указатель на повтор.
    array1[aaddr1++] = 1;
    // В массив адрес первого повтор.
    array1[aaddr1++] = zz;
    // В массив адрес последнего повтор.
    array1[aaddr1++] = z;
    // В массив сколько повтор.
    array1[aaddr1++] = x;
}
else
{
    // Неповторяющиеся.

    // В zz адрес первого неповтор.
    zz = z;

    for (x = 1;;) // Бесконечный цикл.
    {
        // Достигнут конец buffer'a.
        if (z == bl - 1)
        {

```

```

        // Прервать цикл.
        break;
    }

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

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

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

// Для проверки.
// alert("Адрес первого неповтор. " + zz + ". " + "Адрес последнего неповтор. " + z +
// ". " + "Сколько неповтор. " + x + ".");

// В массив указатель на неповтор.
array1[aaddr1++] = 0;
// В массив адрес первого неповтор.
array1[aaddr1++] = zz;
// В массив адрес последнего неповтор.
array1[aaddr1++] = z;
// В массив сколько неповтор.
array1[aaddr1++] = x;
}
z++;
}
}

return array1;
}

```