

```

//
// 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 BuildArrayAddr8(buffer)
{
// buffer - это массив с числами (каждая ячейка от 0 и до 255, байты)!
// buffer - это тот массив, который нужно будет "подсунуть"
// кодировщику (кодировщик см. ниже).
//
// 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 bl, aaddr1, addr2, addr3, total, pl;

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

aaddr1 = 0;

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

addr3 = 0; addr2 = addr3; total = 1;

// В массив указатель на неповтор.
array1[aaddr1++] = 0;
// В массив адрес первого неповтор.
array1[aaddr1++] = addr2;
// В массив адрес последнего неповтор.
array1[aaddr1++] = addr3;
// В массив сколько неповтор.
array1[aaddr1] = total;
}
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)
{
addr2 = addr3; total = 1;
}
}
}
}

```

```

// В массив указатель на неповтор.
array1[aaddr1++] = 0;
// В массив адрес первого неповтор.
array1[aaddr1++] = addr2;
// В массив адрес последнего неповтор.
array1[aaddr1++] = addr3;
// В массив сколько неповтор.
array1[aaddr1] = total;

// Прервать цикл.
break;
}

// Один раз можно выполнить if, а потом сделать через switch.
// Надо подумать!
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'а.
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;
}
}

// В массив указатель на повтор.
array1[aaddr1++] = 1;
// В массив адрес первого повтор.
array1[aaddr1++] = addr2;
// В массив адрес последнего повтор.
array1[aaddr1++] = addr3;
// В массив сколько повтор.
array1[aaddr1++] = total;
}
else
{
// Неповторяющиеся.

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

for (total = 1;;) // Бесконечный цикл.
{
// Достигнут конец buffer'а.
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++;  
    }  
}  
  
// В массив указатель на неповтор.  
array1[aaddr1++] = 0;  
// В массив адрес первого неповтор.  
array1[aaddr1++] = addr2;  
// В массив адрес последнего неповтор.  
array1[aaddr1++] = addr3;  
// В массив сколько неповтор.  
array1[aaddr1++] = total;  
}  
addr3++;  
}  
}  
  
return array1;  
}
```