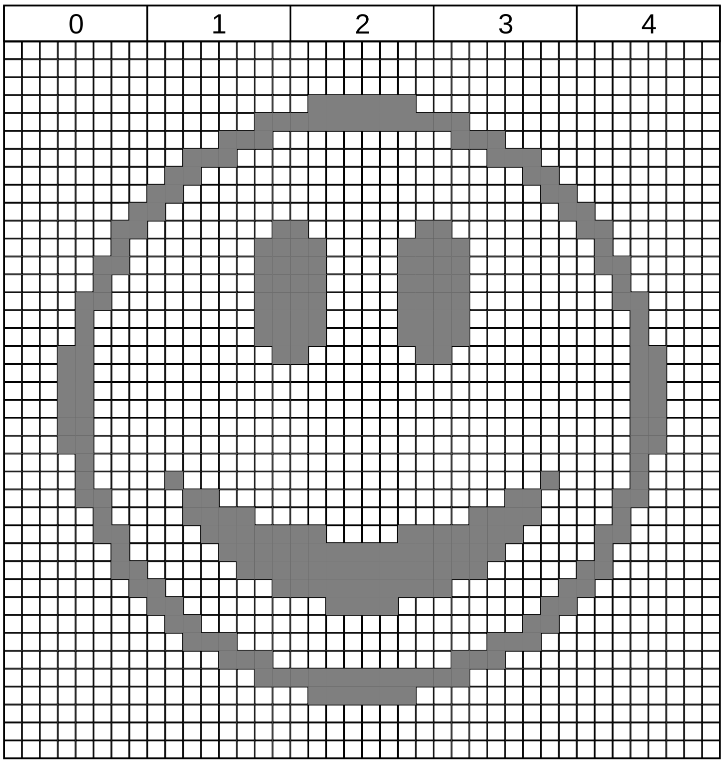
 English name: ____________________

Worksheet: Run-Length Encoding

1. Examine the raster image to the right. The hexadecimal values for the first few rows of the bitmap format pixel array is given below, using the color table values of zero (0) for black and one (1) for white.
   For simplicity, we will not pad the rows to a multiple of 32 bits (4 bytes).



		byte				
		0	1	2	3	4
row	0	FF	FF	FF	FF	FF
	1	FF	FF	FF	FF	FF
	2	FF	FF	FF	FF	FF
	3	FF	FF	7E	FF	FF
	4	FF	FC	00	3F	FF
	5	FF	F1	FF	7F	FF

a) Complete the missing values from the table, above.

b) The image above is 40 pixels high. How many bits are required to store the entire pixel array? ***1600 bits***

c) Convert the number of bits in part (b) to bytes. How many bytes are required to store the entire pixel array? ***200 bytes***

2. Now we will encode the first few rows of the bitmap using ***run-length encoding***. Recall that run-length encoding stores the value of the data, followed by the number of repetitions of that data.

   For our encoding, we will use a single byte, with the first bit representing the color of the pixels, and the remaining 7 bits representing the number of pixels of that color. This encoding is represented in the diagram in the box to the right.

   $\overbrace{0}^{\text{color}}\ \overbrace{0\,0\,0\,1\,0\,1\,0}^{\text{repetitions}}$

a) Count the number of white pixels from the start of the image, traversing each row, left to right, until the first black pixel. How might we represent this using our run-length encoding scheme?
Hint: we will not be able to represent it in a single byte as there are too many repetitions.

Binary: 11111111 10001010 Hexadecimal: 0xFF8A

b) Continue for the next sixteen bytes; write the binary.

00000110	10011111	00001100	10011010
00000011	10001010	00000011	10010110
00000011	10001110	00000011	10010011
00000010	10010010	00000010	10010001

3. Use the compression ratio formula given to the right to complete the following.

$$\textit{compression ratio} = \frac{\textit{original size}}{\textit{compressed size}}$$

a) In part (2a), how many bits of the pixel array were encoded into two bytes (16 bits) using run-length encoding?

137

b) Use the information in part (3a) to calculate the *compression ratio* of the first two bytes of the run-length encoding.

$\frac{137}{16} \approx$ **8.6 : 1**

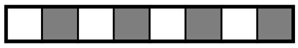
c) In part (2b), how many bits were encoded into the subsequent sixteen bytes (128 bits) of run-length encoding?

328

d) Use the information in part (3c) to calculate the *compression ratio* of the subsequent 16 bytes of the run-length encoding.

$\frac{328}{128} \approx$ **2.56 : 1**

4. Consider the 1x8 pixel two-color image to the right.



a) How many bits would be necessary to encode the image as a pixel array?

8 bits

b) How many bytes would be necessary to encode the image using the above run-length encoding scheme?

$8 \times 8 =$ **64 bits**

c) What is the *compression ratio* when the above run-length encoding scheme is used to compress this image?

$\frac{8}{64} = \frac{1}{8} =$ **0.125 : 1**

5. Answer in complete sentences.

a) What type of data does run-length encoding work well on (provide good compression)?

Run-length encoding works well on data that has long strings of repetition (repeated characters, the same color pixel repeated, etc.)

b) Comment on the characteristics of and the storage space for the 8-pixel image in question 4.

The image does not have any repeated pixels, and the run-length encoding requires even more space than storing the image as a pixel array!

c) Why might run-length encoding not work well on a high-color image (such as those on page 142 of your textbook)?

If one looks closely, one can see that even when the color is almost the same, there is slight variation in the color. Even a slight variation in color means a new value to represent the color, so a new run.

d) Describe how to interpret different ranges of values for the ***compression ratio***.

The larger the compression ratio, the better the compression. A compression ratio of one (1) means that the "compressed" version is exactly the same size as the original. A compression ratio greater than one means the "compressed" data actually takes more space than the original.