ASSIGNMENT WEEK 6

In class we have looked at the PCX picture format to implement run-length encoding. For this assignment you will implement your version of BI_RLE8 – Microsoft's run-length encoding for 8-bit bitmaps. Reading and writing binary data with correct header (RIFF) information is not part of this assignment.

BI_RLE8 is a format for indexed bitmaps with 8 bits per pixel. The compression format is a two-byte format consisting of a non-zero count byte followed by a byte containing a color index, or an "absolute mode".

- Encode mode
  - First byte is nonzero and is the `count` byte.
  - Second byte is the index of the 8-bit color to be repeated `count` times.
- Absolute mode
  - First byte is zero.
    - Second byte is 0x03h (3) through 0xffh (255). The second byte represents the number of bytes that follow, each of which contains the color index of a single pixel.
    - Second byte is 0x00h encodes an end-of-line. **Do not use!**
    - Second byte is 0x01h encodes end of bitmap. **Do not use!**
    - Second byte is 0x02h. The next two bytes are an x-y-offset for the next pixel. **Do not use!**

You will implement two methods as outlined below in **Python**.

0. If your Python environment does not have the (fork of) Python Image Library ***Pillow***, install it first. `pip install pillow` should work in a standard Python environment with correctly working paths. Alternatively, install from within PyCharm when offered to install a missing library. Load the picture provided using the ***Pillow*** library function `Image.open`. Create a list of pixels using `list` and the `getdata` method of ***Pillow***. You should now have a list of indexed colors 0 – 255.
1. Write a function `to_BI_RLE8_EM` which takes a list of integers (indexed colors ranged 0 – 255) and returns a list of run-length encoded values. Implement the "Encode Mode" ***only***.
2. Write a function `to_BI_RLE8` which takes a list of integers (indexed colors ranged 0 – 255) and returns a list of run-length encoded values. Implement it with both the "Encode Mode" and the first case of "Absolute Mode" (which should be used when there are runs of length 1). Within your program you should determine which mode creates a shorter output for the next pixels. Document your reasoning for the choice of mode to select in your code.
3. Test and debug your methods. Provide test runs in form of a main (driver) file. Compare the original size of the picture with the sizes you obtain from 1. and 2.

Each of the problems 1 – 3 will be graded according to the following rubric for a total of 12 points.

SCORE	**4**	**3**	**2**	**1**	**0**
SKILL LEVEL	Response gives evidence of a complete understanding of the problem; is fully developed; is clearly communicated.	Response gives the evidence of a clear understanding of the problem but contains minor errors or is not fully communicated.	Response gives evidence of a reasonable approach but indicates gaps in conceptual understanding. Explanations are incomplete, vague, or muddled.	Response gives some evidence of problem understanding but contains major math or reasoning errors.	No response or response is completely incorrect or irrelevant.