Answer on Question #57700 – Programming & Computer Science - Python

Let p be the number of bytes skipped after the header before the first bit of the hidden message appear.

Let q be the number of bytes skipped between each bit of the hidden message.

Let the name of the image be 'image.bmp'.

Let the file english_sample.txt be the random english text needed to analyse ratio between numbers of 1 and 0 in its binary representation. Note: if we know that the hidden text is in some particular form, e.g. formatted in UPPERCASE, so should be english_sample.txt.

Then the following python code outputs the values of p, q and the hidden text:

```
# code
expected_R = 0.47
with open('randomtextgenerator.com.txt') as f:
    t = f.read()
    expected_R = sum(bin(ord(c)).count('1') for c in t.upper()) / (len(t)*8) tolerance = .1 with
open("image.bmp","rb") as f:
    data = bytearray(file.read())
min_text_len = 10
for p in range(len(data)):
    text_is_found = True
    max_q = (len(data)-p) // (min_text_len*8)
    for q in range(1, max_q):
        text_is_found = True
        header_len = 54
        text = bytearray((len(data) - header_len - p) // (q*8))
        count_of_1 = 0
        for i in range(len(text)):
            text[i] = 0
            for j in range(8):
                bit = (data[header_len + p + (i * 8 + j) * q] & 0b00000001)
                count_of_1 += bit
                text[i] = text[i] | (bit << j)
            if int(text[i]) > 127: # verify that characters are legal
                text_is_found = False
                break
        R = count_of_1 / (len(text)*8)
        if abs(R - expected_R) < tolerance: # statistical check
            text_is_found = True
            break
    if text_is_found:
        break
print('p, q = {}, {}'.format(p, q))
print(text.decode('ascii'))
# end of code
```