For problem 1, we need to write a script using DES algorithm in both encryption and decryption. Page 21 from lecture note 3 shows us the structure of how DES works. So, I combined and clean the unnecessary code from the giving files into my DES script. After reading the code, we could find out that, we need to write our own code after the step of the first xor. As a result, I did the application of substitution and permutation. After that, I xor the result and LE to find out our RE. We needed to repeat this process 16 rounds to get our final encrypted code (from lecture note page 13). For decryption, I basically did the similar process, but with reversed round key. Since the input now is a very long hex, we need to divide it into several part with length of 64. So, we can read and divide properly. After running through the whole process, I changed the result back into ascii form.

The result for encryption:



The result for decryption:

In the unforgiving world of Formula One, Lewis Hamilton abides at the top. He's the man to beat, the top earner, the most important voice, the most prominent figure - a Black man alone at the summit of motorsports' highest echelon. England's knight in Mercedes armor. Over the past 15 years, the 36-year-old Briton has won seven world championships, tying the record set by Ferrari's Michael Schumacher - the German F1 driver who was regarded as the greatest of all time until Hamilton broadsided him from that perch. At Sunday's Russian Grand Prix, Hamilton rallied through a late rain shower to claim the checkered flag on the way to becoming the first driver in the sport's history with 100 career victories. And that's besides his 100 career pole positions. As achievements go in racing, this is beyond otherworldly.

For problem 2, we need to encrypt the ppm image. This is basically the similar process as encryption for text. However, we need to read off the first 3 lines of the input in rb mode. And, we combine those lines with the output after encryption.

The output for problem 2:

A picture containing shape

Description automatically generated