Problem 1

* I have using some of the lecture notes as starting of my code in def \_\_init\_\_(self, key), including s-boxes, key permutation, p-boxes and etc. Then I followed the page 21 on the Lecture 3 with Feistel Structure to finish up the encrypt function. My code basically went through the steps of divided into left and right block, expansion permutation, xor with round key, substitution with s-box, permutation with p-box, xor with left block, and then doing all over again in the while loop. For the decryption I first change the bitvector of extracting the consecutive 64-bit chunks instead of calling more\_to\_read. I then simply modified certain part of the encryption part such as reverse the order of the round key generated: final\_key = list(reversed(round\_key)), and change the way of reading the blocks. Also at the end I change the final string from reading as hex to ascii.

Encrypted output:



Decrypted output:

Scuderia Ferrari is the racing division of luxury Italian auto manufacturer Ferrari and the racing team that competes in Formula One racing. The team is also known by the nickname "The Prancing Horse", in reference to their logo. It is the oldest surviving and most successful Formula One team, having competed in every world championship since the 1950 Formula One season. The team was founded by Enzo Ferrari, initially to race cars produced by Alfa Romeo. By 1947 Ferrari had begun building its own cars. Among its important achievements outside Formula One are winning the World Sportscar Championship, 24 Hours of Le Mans, 24 Hours of Spa, 24 Hours of Daytona, 12 Hours of Sebring, Bathurst 12 Hour, races for Grand tourer cars and racing on road courses of the Targa Florio, the Mille Miglia and the Carrera Panamericana. The team is also known for its passionate support base, known as the tifosi. The Italian Grand Prix at Monza is regarded as the team's home race.

Problem 2

* For the problem 2 I basically copied the encrypt function in problem 1 and changed the way of reading the file and writing the output file to read and write binary form, and keep the first 3 line not encrypted. Then I basically did the same thing change use write\_to\_file to save the output as a ppm file. Below is the output

Problem 2 output:

* A screen shot of a television screen

  Description automatically generated