Image encryption:   
  
I first copied my encrypt function from the last homework and changed it to encrypt\_block(). It now only takes a plaintext bitvector as an input. I moved key initialization to the init() function and that sped up my implementation considerably.

In my ctr\_aes\_image() function, I open the plaintext image as a bitvector until I get 3 newline characters to know that I get through the header. Then I start a while(more to read) loop and read from the plain image 128 bits at a time, padding if the block is ever less than 128 bits. Then I encrypt the block, increment the initialization vector, xor the current chunk with the output of the encrypt\_block function, and then write that result to the output file.

Code:

A screenshot of a computer program

Description automatically generated

Output image:   
A screen shot of a television

Description automatically generated

Random number generation:

Following the diagram from the 10.6 lecture notes, I generated 5 pseudorandom numbers. I used 2 lists for the seeds and the random numbers. In a for loop iterating from 0 to totalNum, I generate an output from the encoded datetime XORed with seed[i], then I generate a generate a second output from datetime XORed with the random number that was generated for that iteration, then I make the next seed by encrypting that output. This function uses the same encrypt\_block() function from the image encryption portion of this assignment. Then I write those 5 random numbers to the output file.

Numbers generated:

A screenshot of a computer

Description automatically generated