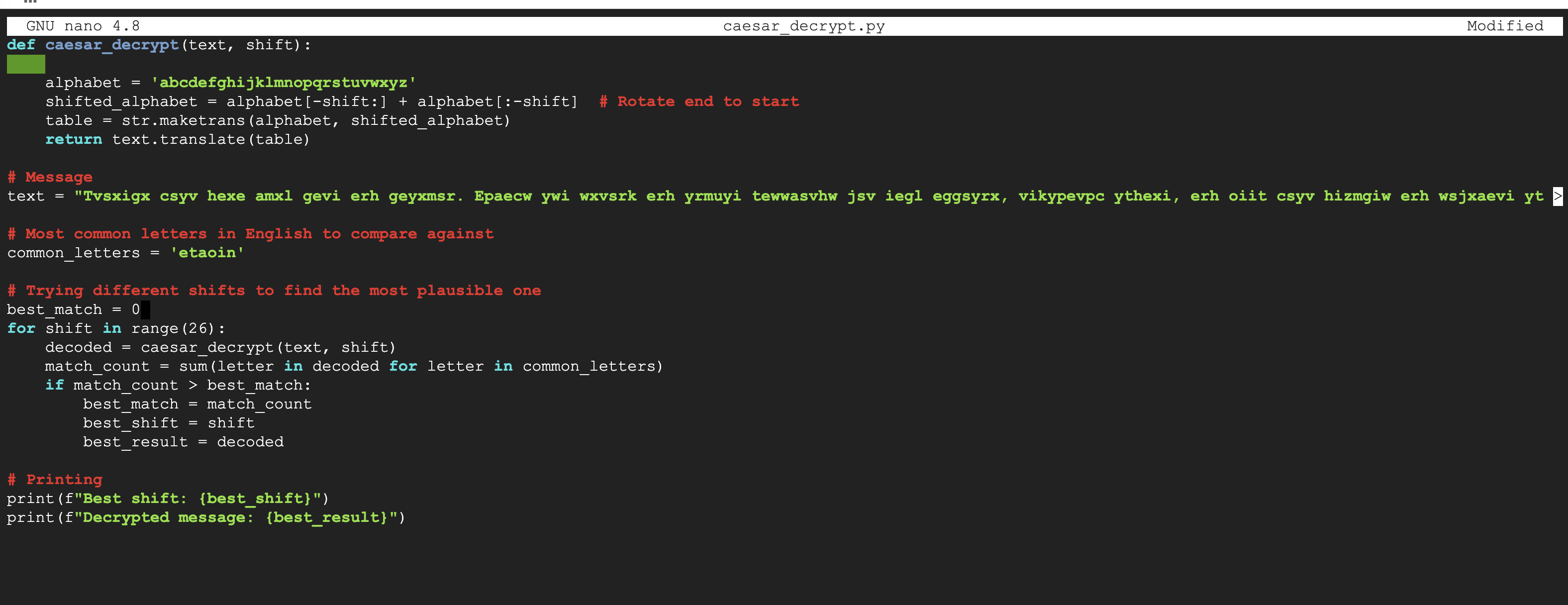
**Name:** Henry Salgado

**ID**: 80509684

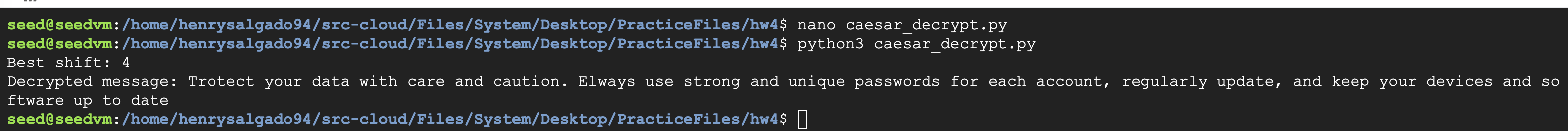
**CS 4351/5352: Computer Security**

**Assignment 4**

**Problem 1: Caesar Cipher**

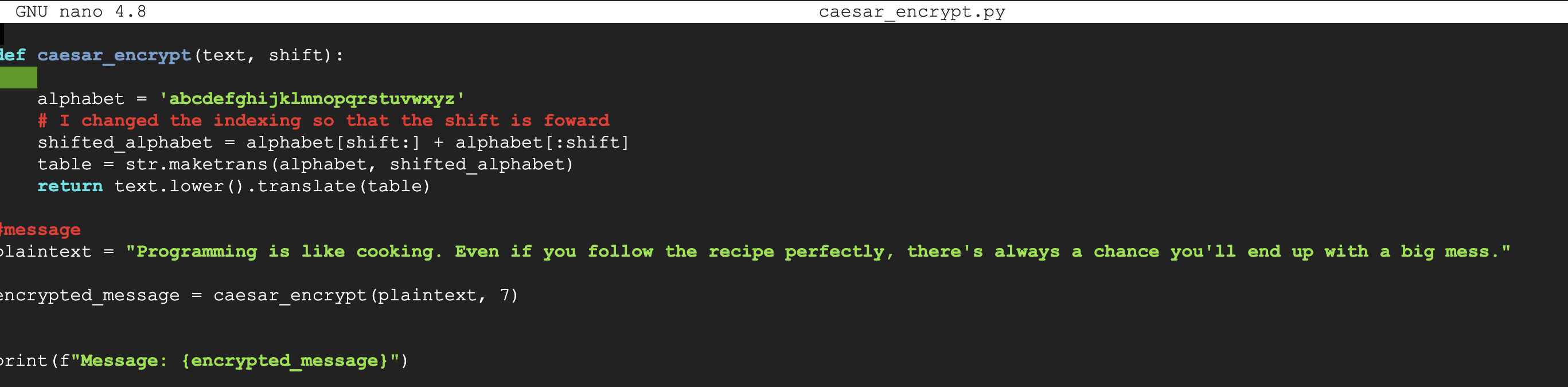


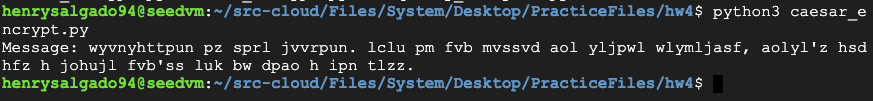
**Running the file:**



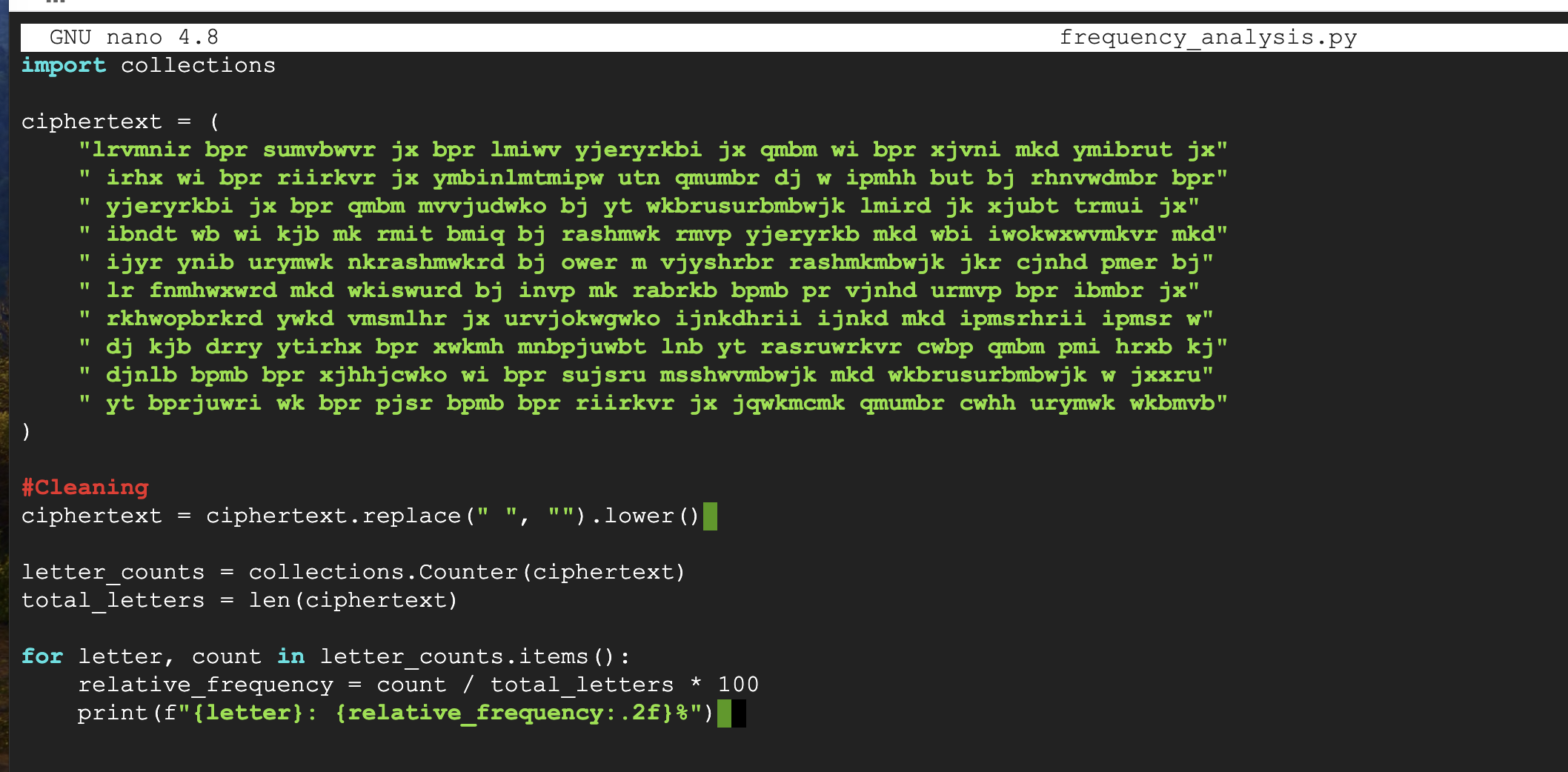
Observations: For the most part, the message is decrypted by my code, finding the best shift to be 4 and the message to be: *“Protect your data with care and caution. Always use strong and unique passwords for each account, regularly update, and keep your devices and software up to date”*

For part 2, I keep the code logic, but change the indexing

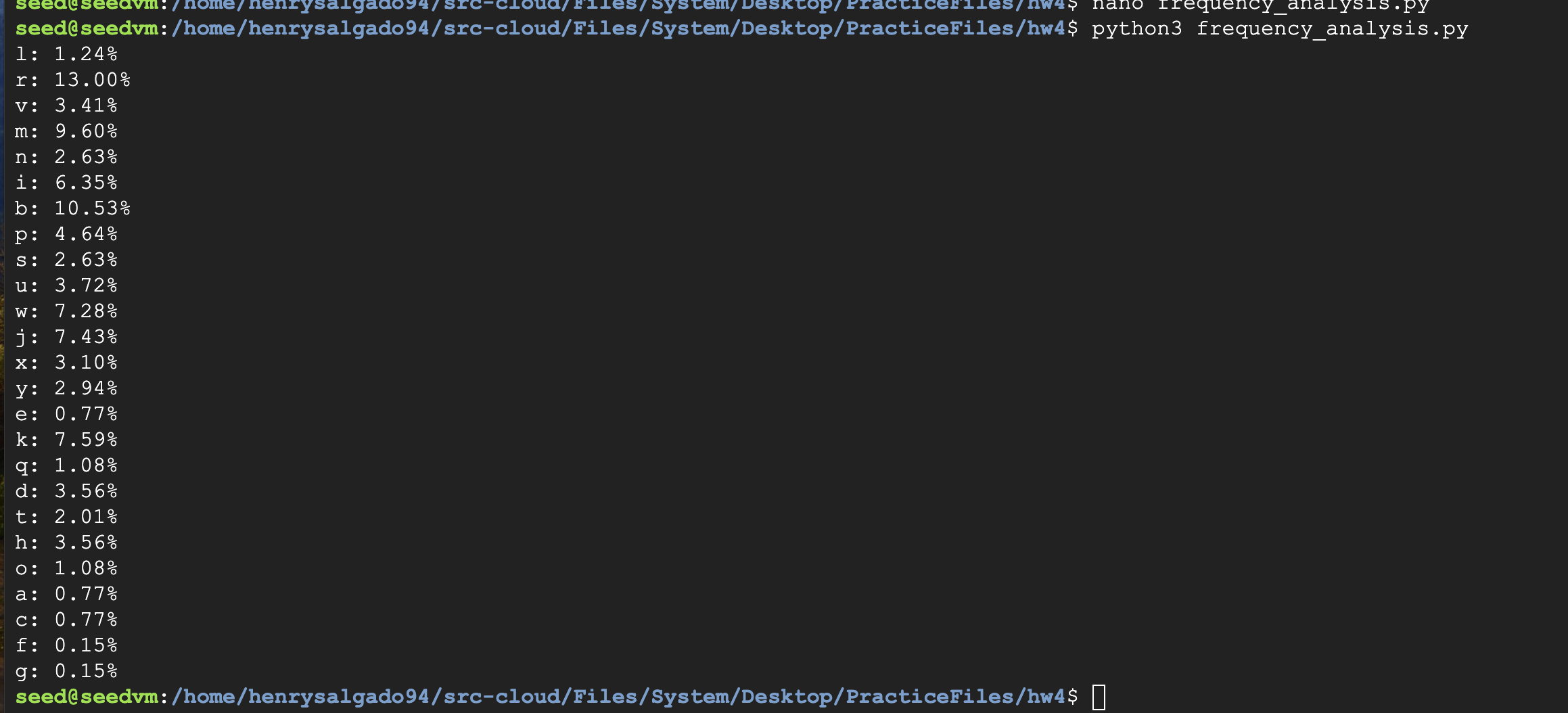




**Task 2: Frequent Analysis**



After compiling and running



According to Google,

* E: Appears most frequently, around 12-13% of the time in typical text.
* T: Second most common, around 9% frequency.
* A, O, I, N: Follow closely behind, all with roughly 7-8% frequencies.
  + Therefore, I will first try r = e

Observation 1: “bpr” appears in the text several times together, and since r = e, I will assume that “bpr” is “the” and if so then b = t and p = h and r = e

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|  | t |  |  |  |  |  |  |  |  |  |  |  |  |  | h |  | e |  |  |  |  |  |  |  |  |

Observation 2: Another common letter is “a” and I have a suspicion that it might match with “m” (9.60%). I see that the text has alot of “mkd” and “mk”. In that case, “mkd” could = “and” and “mk” = “an”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| m | t |  | d |  |  |  |  |  |  | n |  |  |  |  | h |  | e |  |  |  |  |  |  |  |  |

Observation 3: Another common letters in English are “i,o, n” and in my frequencies, I can see that “j” and “w” are around 8%. When searching my text, I see that there are alot of “jx” and “wi”. I am thinking these could be transition words such as “is” or “to” or “of”. I will try “wi” as “is” and “jx” as “of”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| m | t |  | d |  |  |  |  | s | o | n |  |  |  |  | h |  | e |  |  |  |  | i | f |  |  |

Observation 4: There is a sequence “RMIT” = EAS\_. I belive that letter t = y. In that case, the word “RMIT” = “EASY”. There are also two other letter words “YT” and if t = y, then Y = m. Therefore, “YT” = “MY”.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| m | t |  | d |  |  |  |  | s | o | n |  |  |  |  | h |  | e |  | y |  |  | i | f | m |  |

Observation 5: The last word of the paragraph is “wkbmvb” which would translate to “inta\_t”. From the list of english, it could be the word “intact”. If that is the case then v = c.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |
| m | t |  | d |  |  |  |  | s | o | n |  |  |  |  | h |  | e |  | y |  | c | i | f | m |

Observation 6: Second to last word is “urymwk” which would translate to “\_emain”. If the last word was “intact” then a possible previous word could be “remain”. In that case u = r

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |
| m | t |  | d |  |  |  |  | s | o | n |  |  |  |  | h |  | e |  | y | r | c | i | f | m |

Observation 7: The third to last word is “cwhh”. This could translate to \_i\_ \_. It is an insteristing word since it has two repeated HHs. In english, there are not that many repeated letters, and they are often “LL , SS , TT and etc. Considering that the we found “remain intact”, I will this word could be “will”. In that case, c = w and h = L

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |
| m | t | w | d |  |  |  | l | s | o | n |  |  |  |  | h | k | e |  | y | r | c | i | f | m |

Observation 9: The fourth to last word is “qmumbr”, this could be \_arate. I had to think about this one a lot. Since the word should have consonants, one of the words could fit would “KARATE”. In that case, q = K

Observation 10: I will go back to the top of the paragraph and began translating from there.

* The first word is “lrvmnir” = \_eca\_se -> Because. L = b and n = u

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |
| m | t | w | d |  |  |  | l | s | o | n | b |  | u |  | h | k | e |  | y | r | c | i | f | m |

Observation 11: Second word is bpr = the, third word is “sumvbwvr” = \_ractice -> Practice. Therefore, s = p.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |
| m | t | w | d |  |  |  | l | s | o | n | b |  | u |  | h | k | e | p | y | r | c | i | f | m |

Observation 12: The fifth word in the paragraph is “lmiwv” = basic and the sixth word is “yjeryrkbi” = mo\_ement -> movement. Therefore, e = v

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |
| m | t | w | d | v |  |  | l | s | o | n | b |  | u |  | h | k | e | p | y | r | c | i | f | m |

With these we can determine the sentence as *“Because the practice of the basic movements of kata is the focus and mastery of self is the essence of matsubayashi ryu karate do I shall try to elucidate the movements of the kata according to my interpretation based on forty years of study it is not an easy task to explain each movement and its significance and some must remain unexplained to give a complete explanation one would have to be qualified and inspired to such an extent that he could reach the state of enlightened mind capable of recognizing soundless sound and shapeless shape i do not deem myself the final authority but my experience with kata has left no doubt that the following is the proper application and interpretation i offer my theories in the hope that the essence of okinawan karate will remain intact.*

**Question 3: DES (DAT Encryption Standard)**

* **Block Cipher:** DES operates on fixed-size data blocks (64 bits in this case). It encrypts each block independently using a secret key.
* **Key Length: The key length in DES is 56 bits, but it starts with a 64-bit key.** Certain bits are discarded for parity.
* **Number of Rounds:** The encryption process in DES consists of 16 rounds of mathematical operations on the data block and the key.

**Encryption Process:**

1. The permuted block is split into a left half (L0) and a right half (R0).
2. For 16 rounds, each block goes through a function that includes expansion, substitution, and permutation operations. The round function is applied to the right half of the block and the result is XORed with the left half.
3. The output of the XOR is then used as the new right half, and the previous right half becomes the new left half for the next round.
4. After the 16 rounds are completed, the halves are swapped and the combined block goes through a final permutation, which is the inverse of the initial permutation.

**For Decryption of DES:**

* The ciphertext is processed with the same initial and final permutations as encryption.
* The 16 rounds are applied in reverse order, using the subkeys in the reverse order as well.

**Question 4: AES (Advanced Encryption Standard)**

**Block Cipher Size:** 128 bits

**Key Lengths:** AES supports multiple key lengths: 128 bits, 192 bits, and 256 bits.

**Number of Rounds:** The number of rounds in AES depends on the key length. There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys.

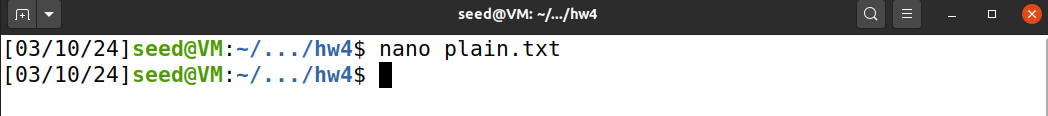
**Encryption Process:**

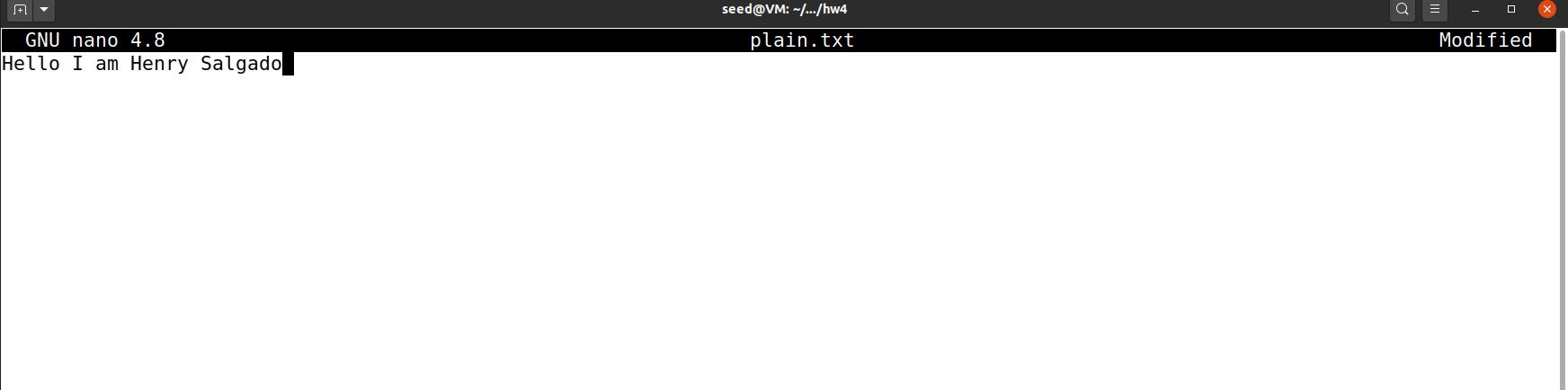
1. Initial Permutation: 128-bit data block is rearranged using a specific substitution table.
2. SubBytes: Each byte (8 bits) in the data block is substituted using a non-linear substitution table.
3. ShiftRows: The rows of the data block are shifted by a certain number of positions to disrupt data organization.
4. MixColumns: The columns of the data block are mixed using a specific mathematical operation to further enhance diffusion.
5. AddRoundKey: The data block is XORed with a round key derived from the main key. This is where the key gets incorporated into the encryption process.
6. Rounds 2 to (N-1): Steps 2-5 are repeated for a specified number of rounds (depending on the key length). In each round, a different round key is used.
7. Final Round: The final round consists of steps 2, 3, and 4, but without the MixColumns step.
8. Final Permutation: The data block is subjected to a final permutation using another substitution table.

**Note:** My Google VM stopped working so I had to go back and download the VM on my machine.

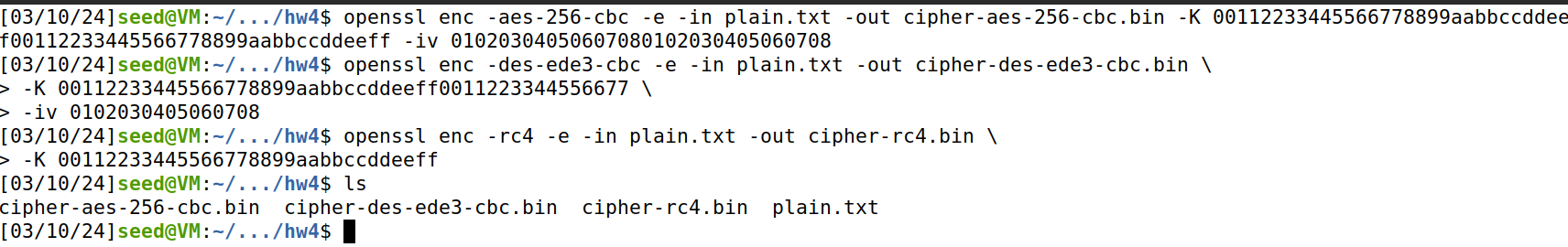
**Question 5:**

First, I will create a plain text file that will serve as input for my encryption

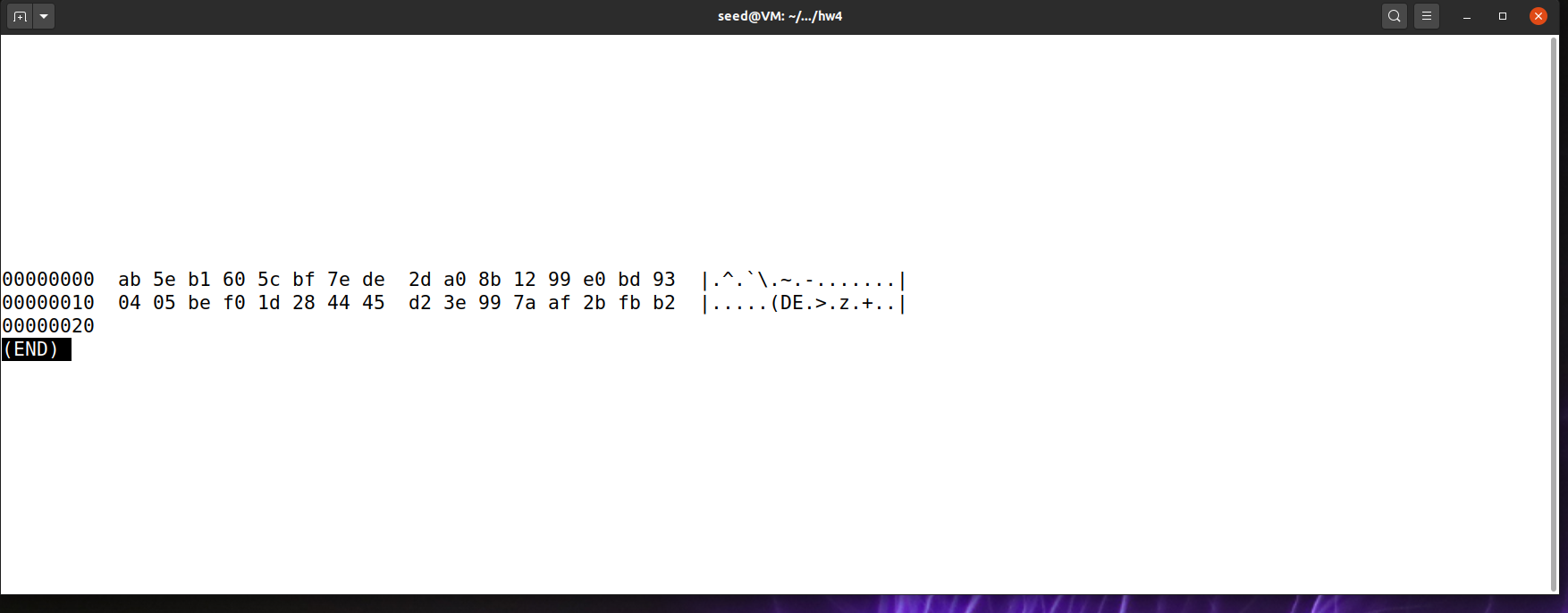




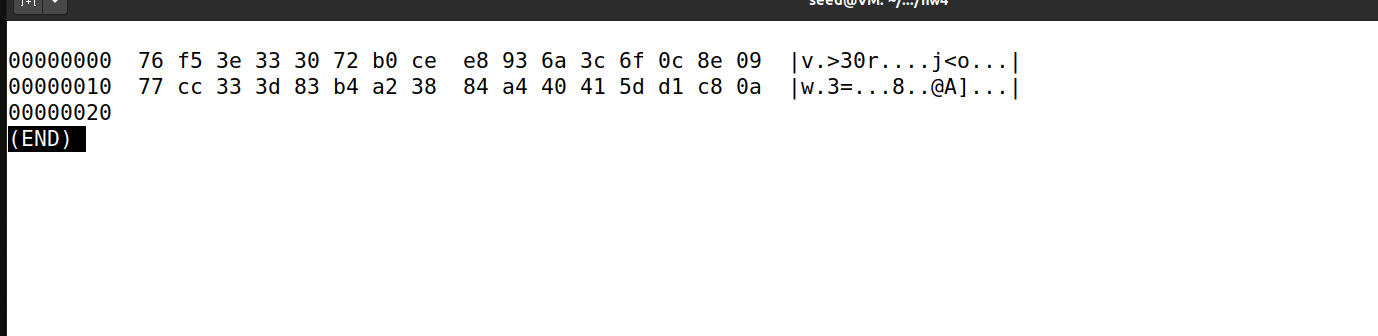
I will try three different ciphers 1) AES-256-CBC 2) DES-EDE3-CBC 3) RC4

**Outputs**

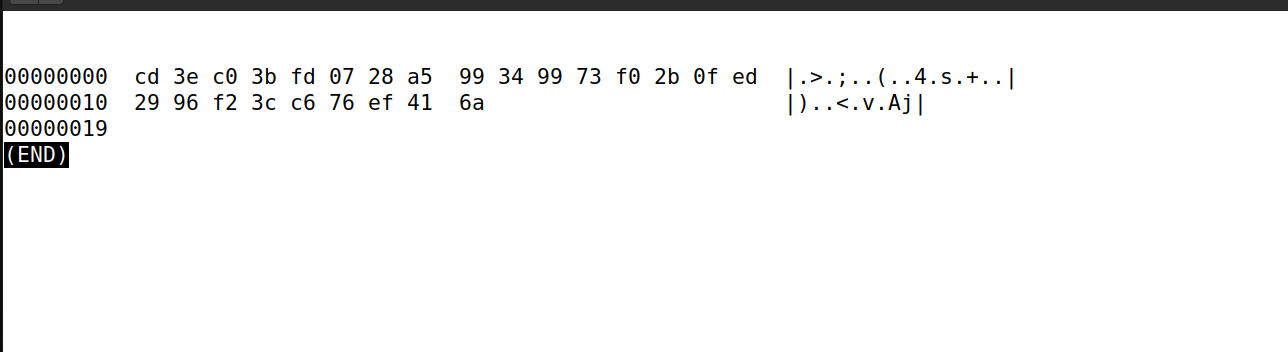




hexdump -C cipher-des-ede3-cbc.bin | less

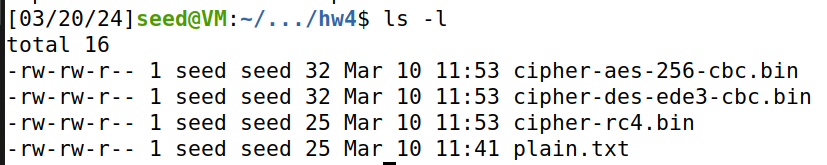


hexdump -C cipher-rc4.bin | less



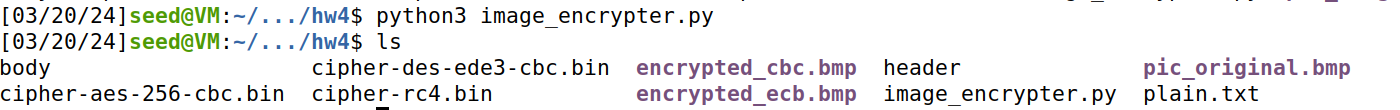
**Observations:**

* No clear patterns in hex output
* Both AES-256-CBC and DES-EDE3-CBC encrypted files are of the same size (32 bytes). This uniformity in size might be because these are both block ciphers.
* The encrypted file using RC4 is slightly smaller (25 bytes) than those encrypted with the block ciphers. RC4 is a stream cipher, meaning it encrypts data one bit or byte at a time and does not require padding to align with a block size.

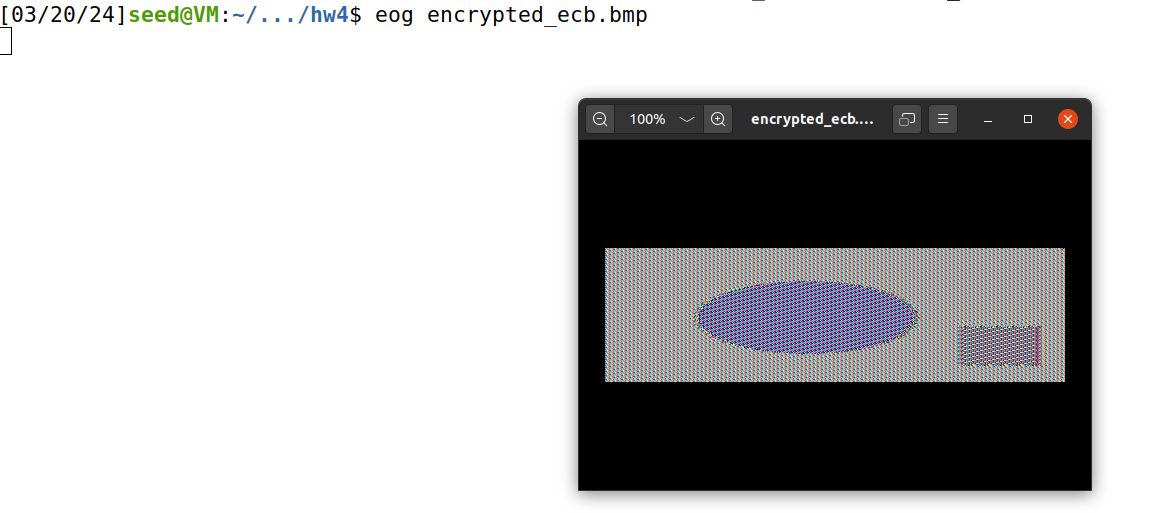


**Question 6:**

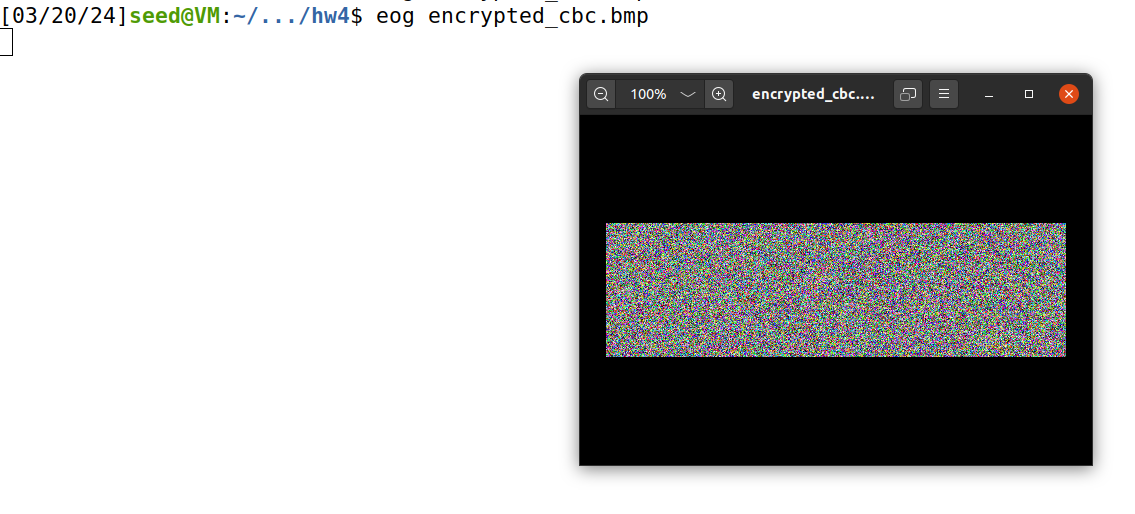
Instead of using the provided commands for combining header and body, I will put commands in a python code called image\_encrypter.py. This will generate both cbc and ecb images for me.



Then, I will view the images using eog



Now, cbc



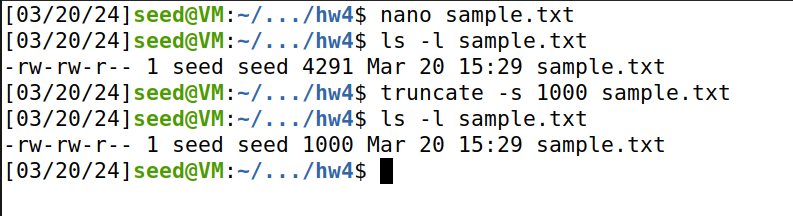
**Observations:**

ECB Mode: The encrypted image displays some patterns from the original image. This is because identical blocks of plaintext result in identical blocks of ciphertext. This can leak information about the image structure.

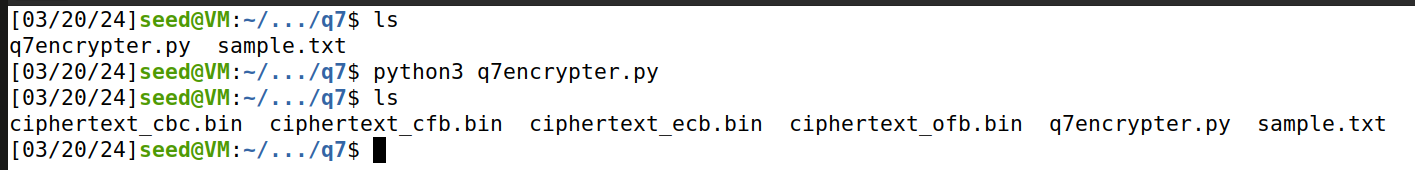
CBC Mode: The encrypted image appears as random noise.

**Question 7:**

First, I put some text from a book in a “sample.txt” file. I checked to see how many bytes, and then truncated to 1000.



I will create a python script that encrypts my sample.txt file using ecb, cbc, cfb, ofb



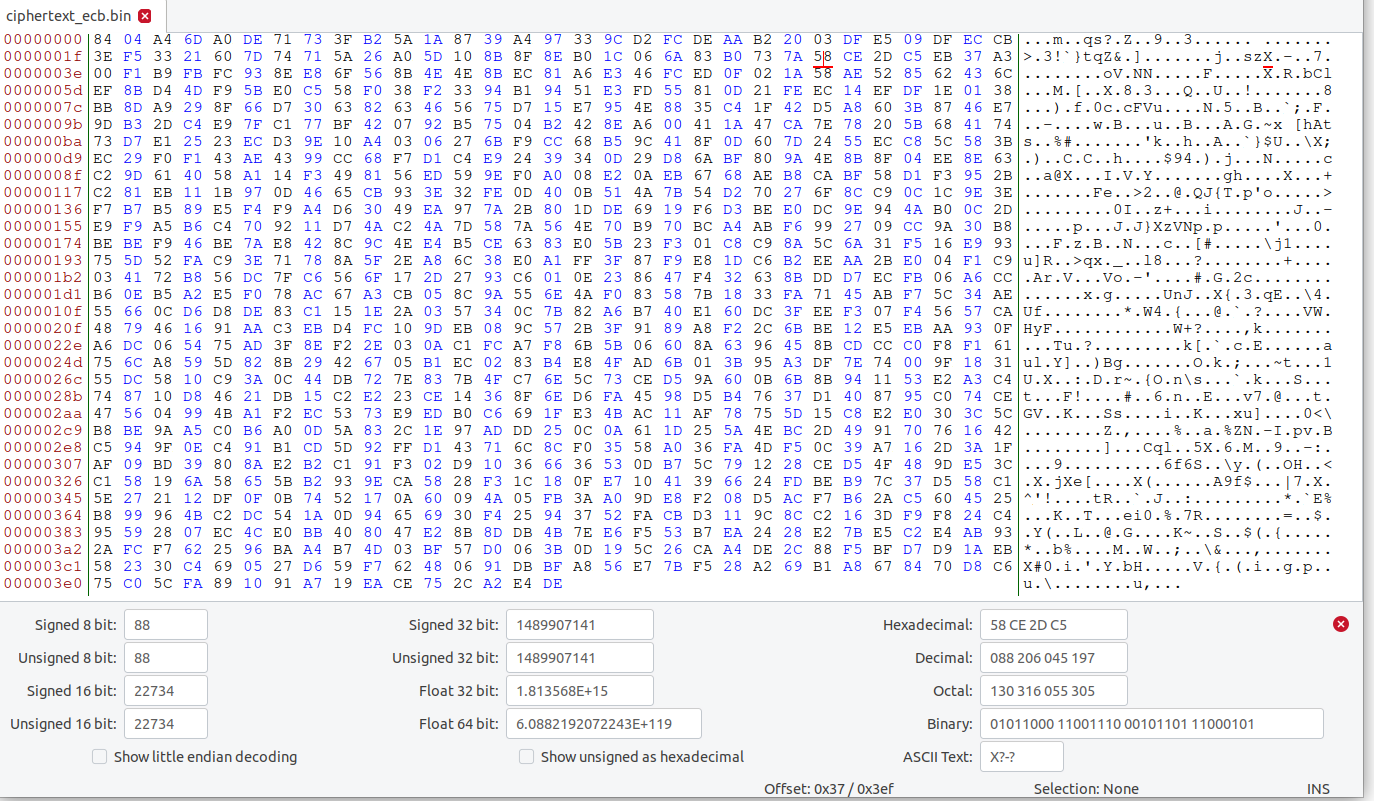
I will open each of the files and corrupt the 55th byte (offset = 0x37)

bless ciphertext\_ecb.bin

bless ciphertext\_cbc.bin

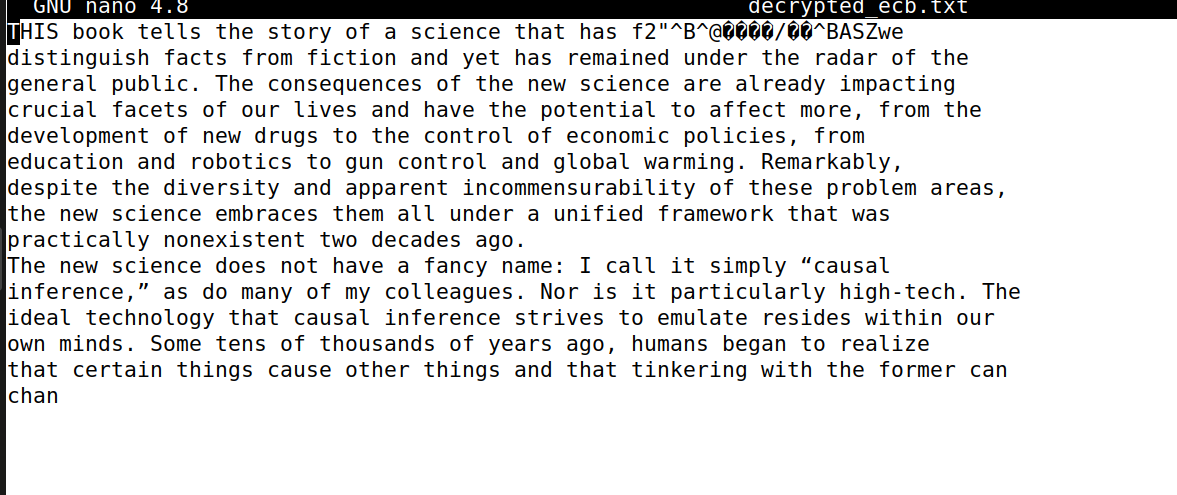
bless ciphertext\_cfb.bin

bless ciphertext\_ofb.bin

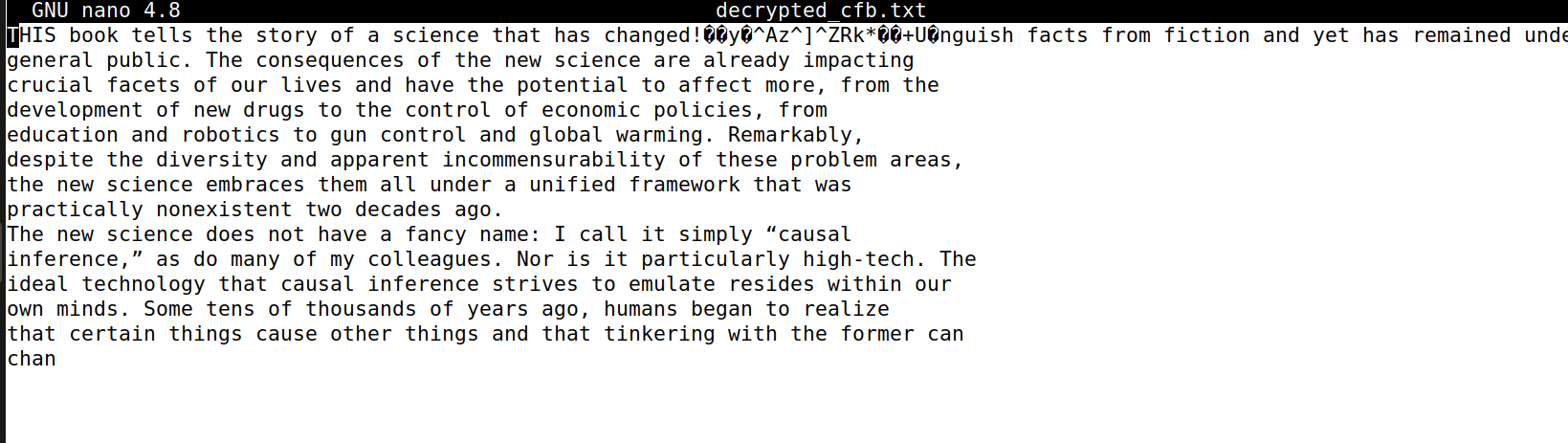


For example, in ecb, I corrupted by changing value from 58 to 57

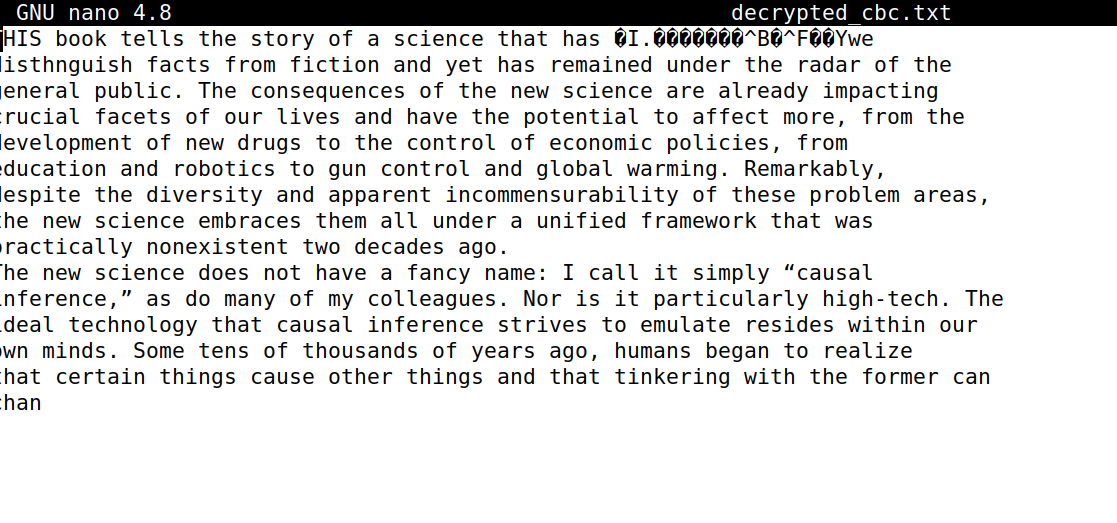
Now, after corrupting all files, I will run de the decrypter and observe the text files. We will start with ECB.



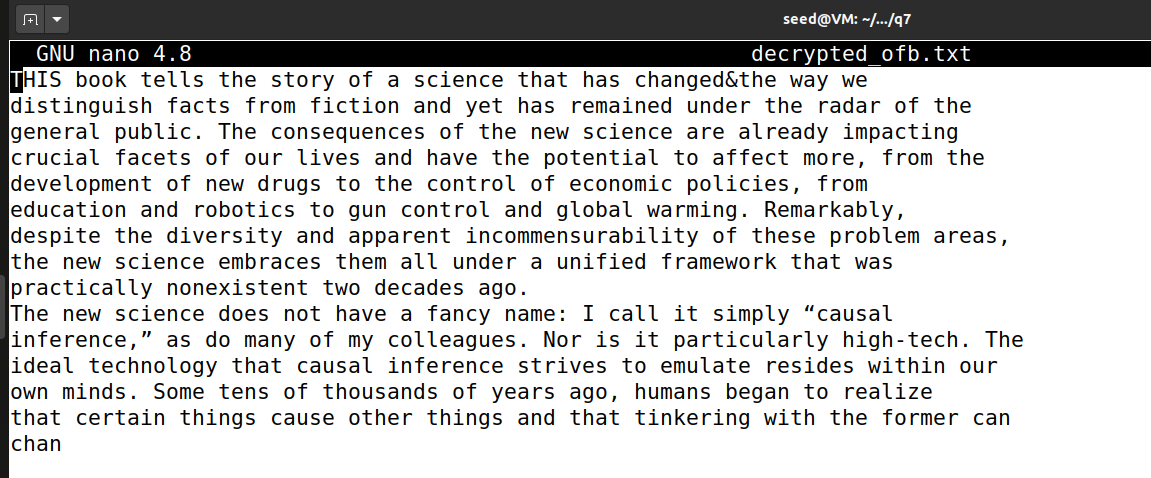
CFB



CBC



OFB



Observations:

ECB (Electronic Codebook) mode:

* In my analysis of ECB mode, I found that each block is encrypted independently, which means that the corruption will only affect the block containing the 55th byte. Upon decrypting the file, I observed that the first 6 blocks (48 bytes) remained intact, while the 7th block (16 bytes) was corrupted. The remaining blocks were decrypted correctly.

CBC (Cipher Block Chaining) mode:

* For CBC mode, I discovered that each block's encryption depends on the previous block's ciphertext. When I introduced corruption in the 55th byte, it caused the 7th block to be decrypted incorrectly, and this error propagated to the 8th block as well. As a result, the decrypted file had the first 6 blocks (48 bytes) intact, but the 7th and 8th blocks (32 bytes) were corrupted. The remaining blocks were decrypted correctly.

Note: Very similar results for CFB

OFB (Output Feedback) modes:

* For OFB modes, I learned that the encryption of each block depends on the previous block's keystream. When I introduced corruption in the 55th byte, it caused a single byte error in the decrypted file at the same position. Consequently, the decrypted file had a single byte corrupted at the 55th position, while the rest of the file was decrypted correctly.