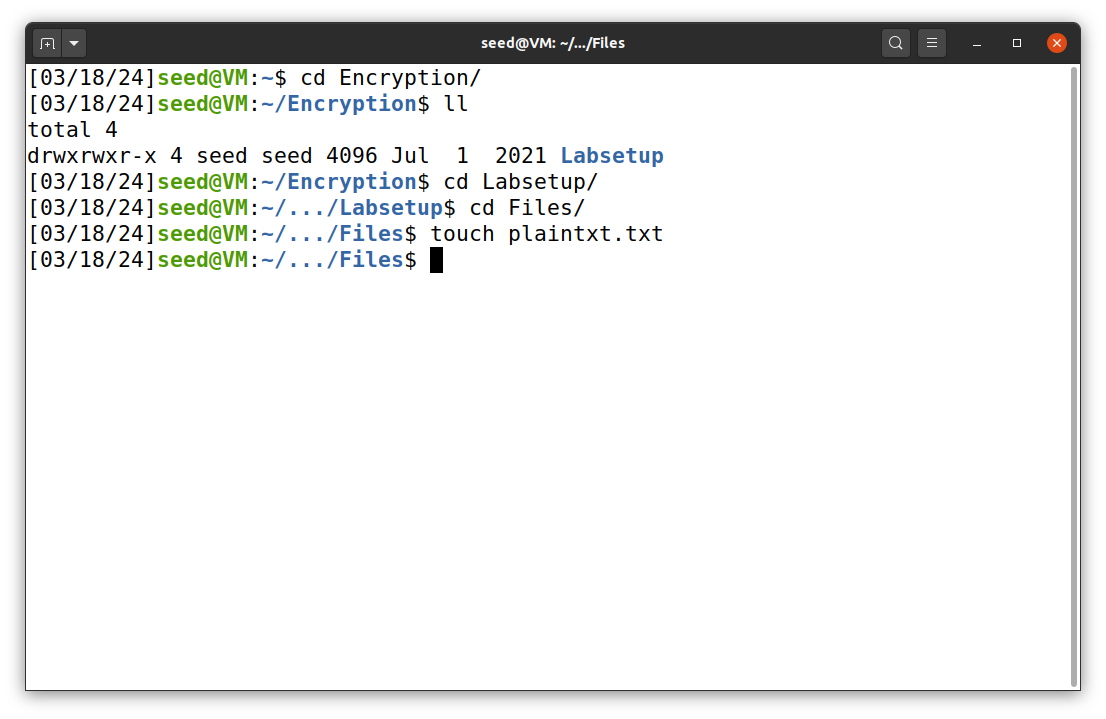
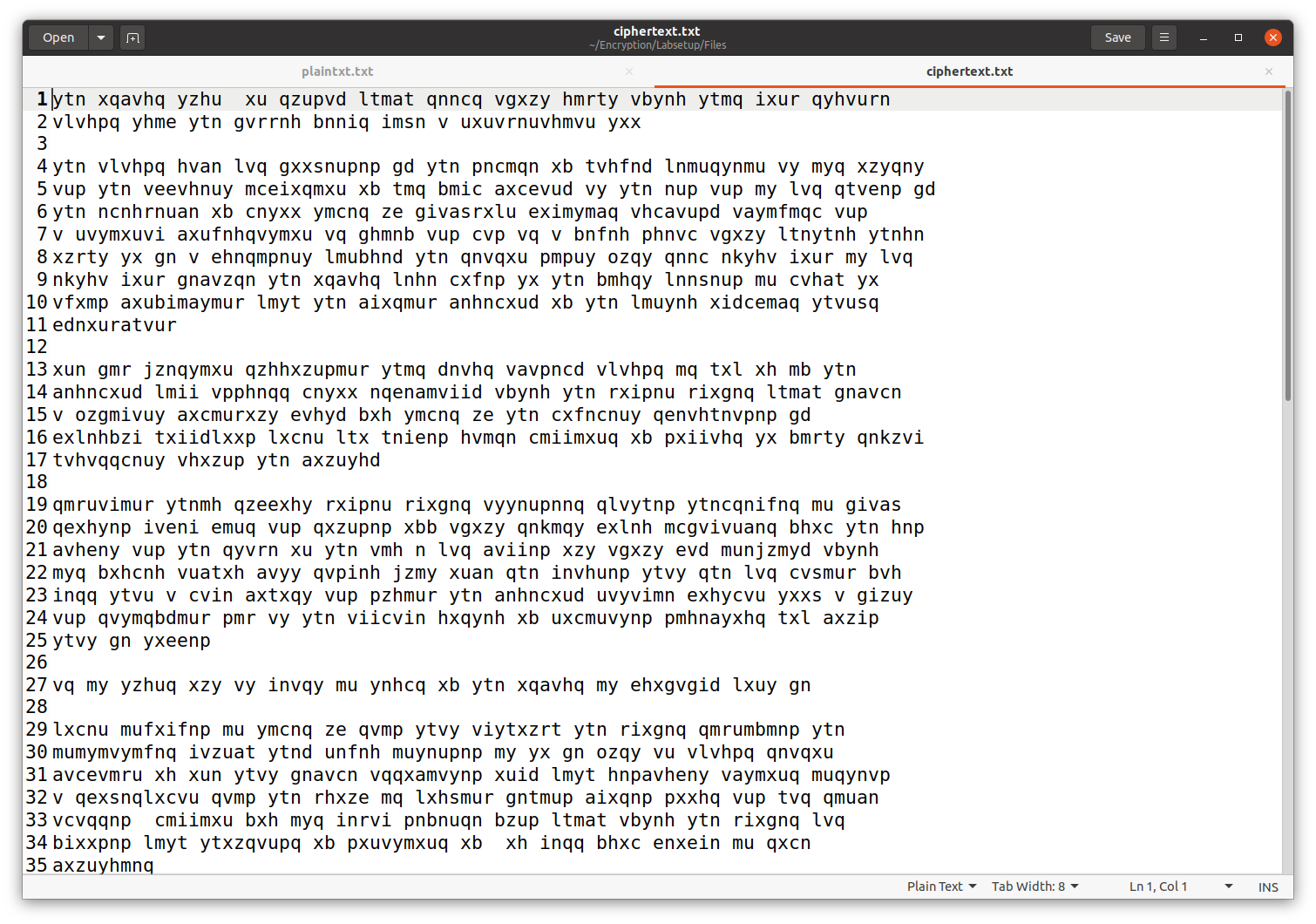
**Lab 8 Secret-Key Encryption**

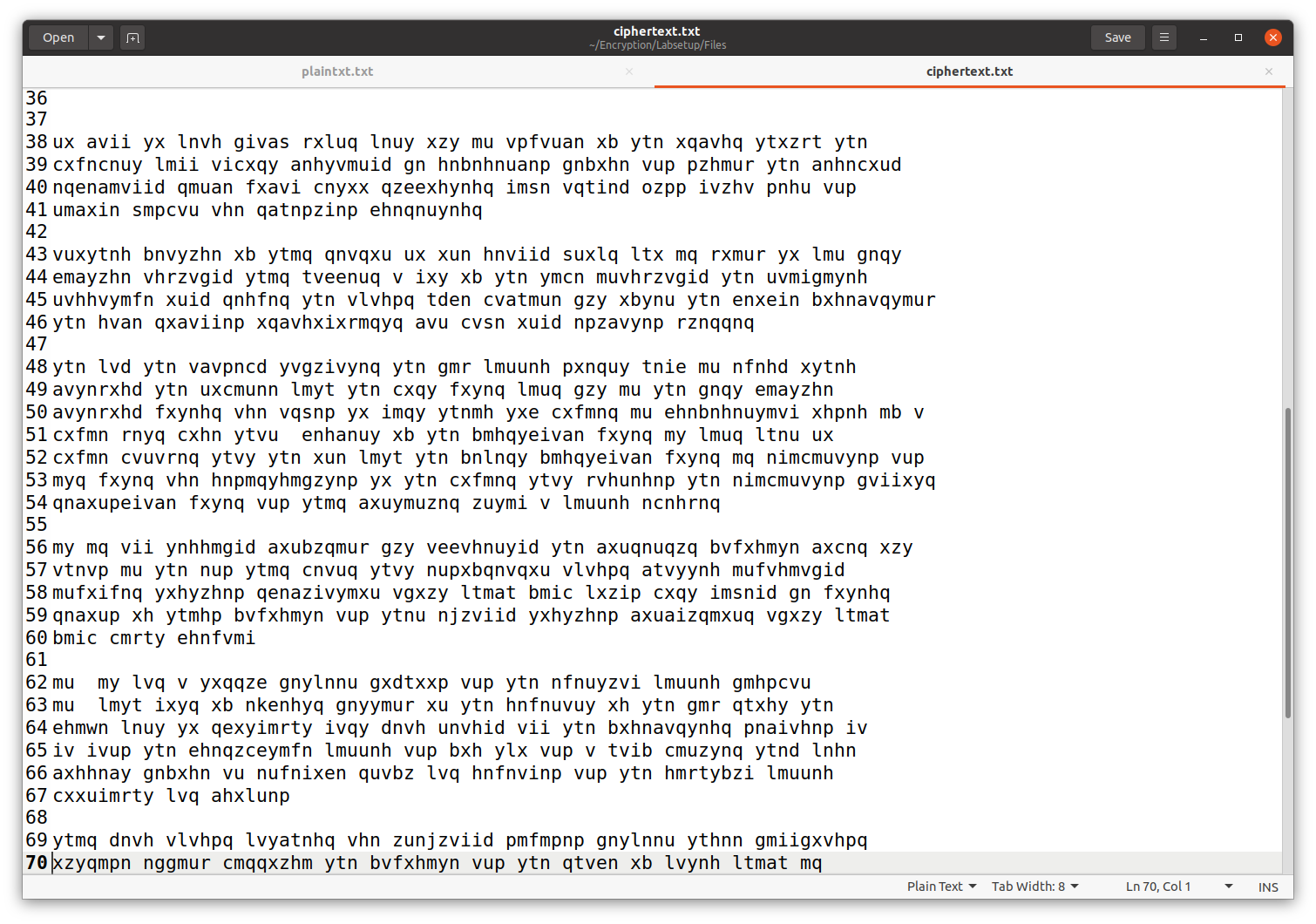
**Task 1: Frequency Analysis**

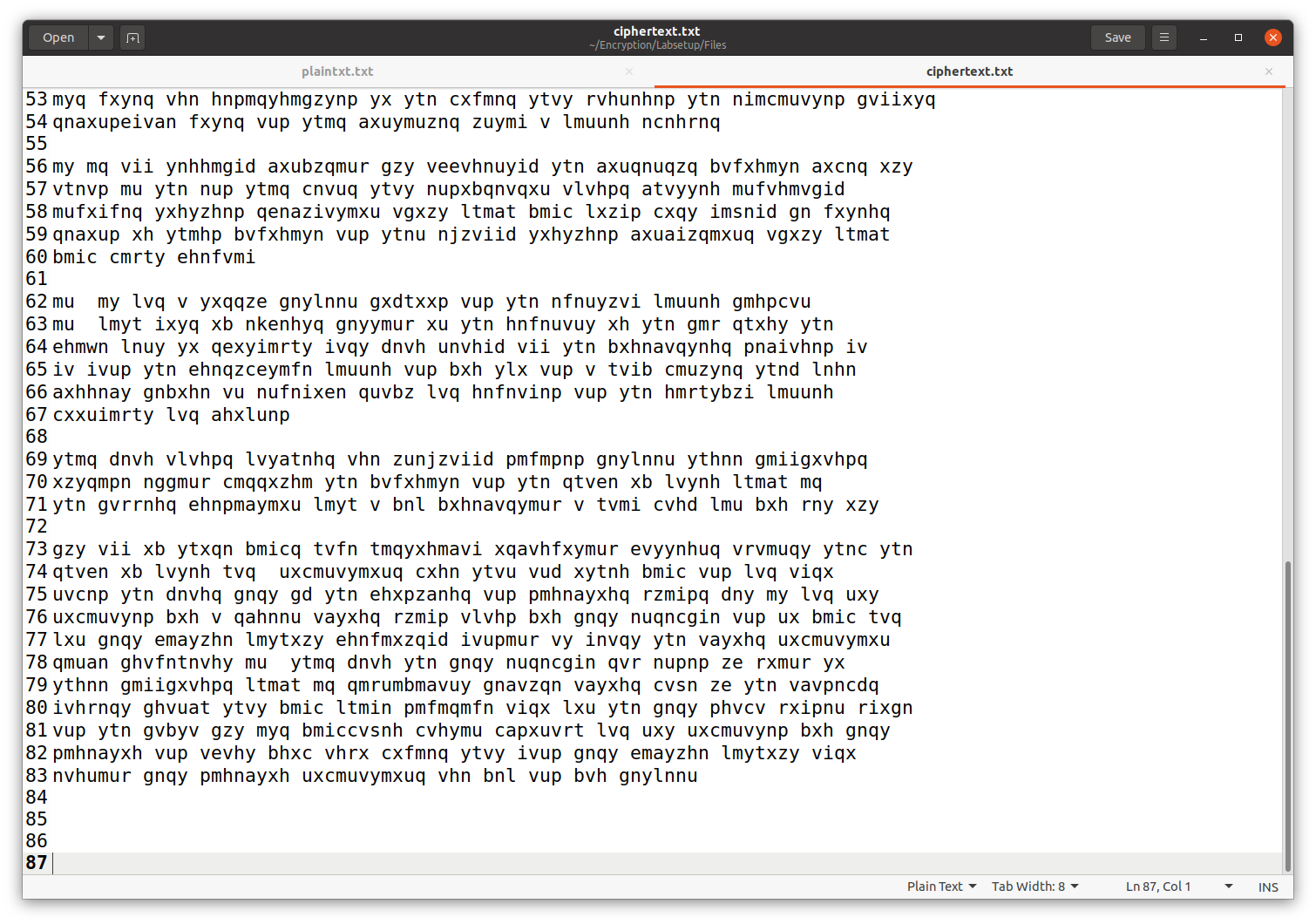
Firstly, I created a new text file called plaintxt.txt to store the deciphered cipher text.

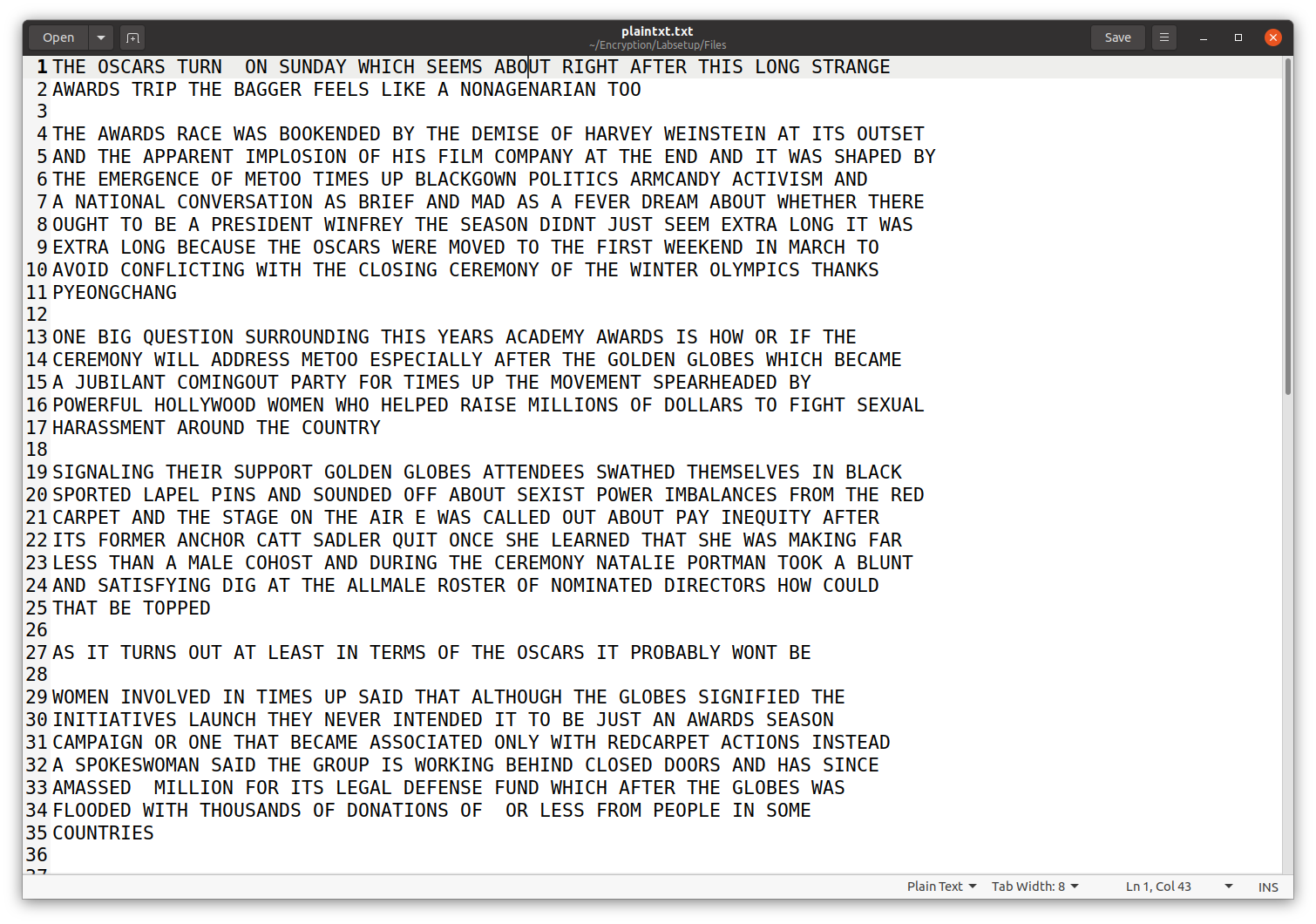
Then I used the watch command on the plain text file – the command I used is – “watch -n 2 cat plaintxt.txt”. So after every 2 intervals this will refresh.

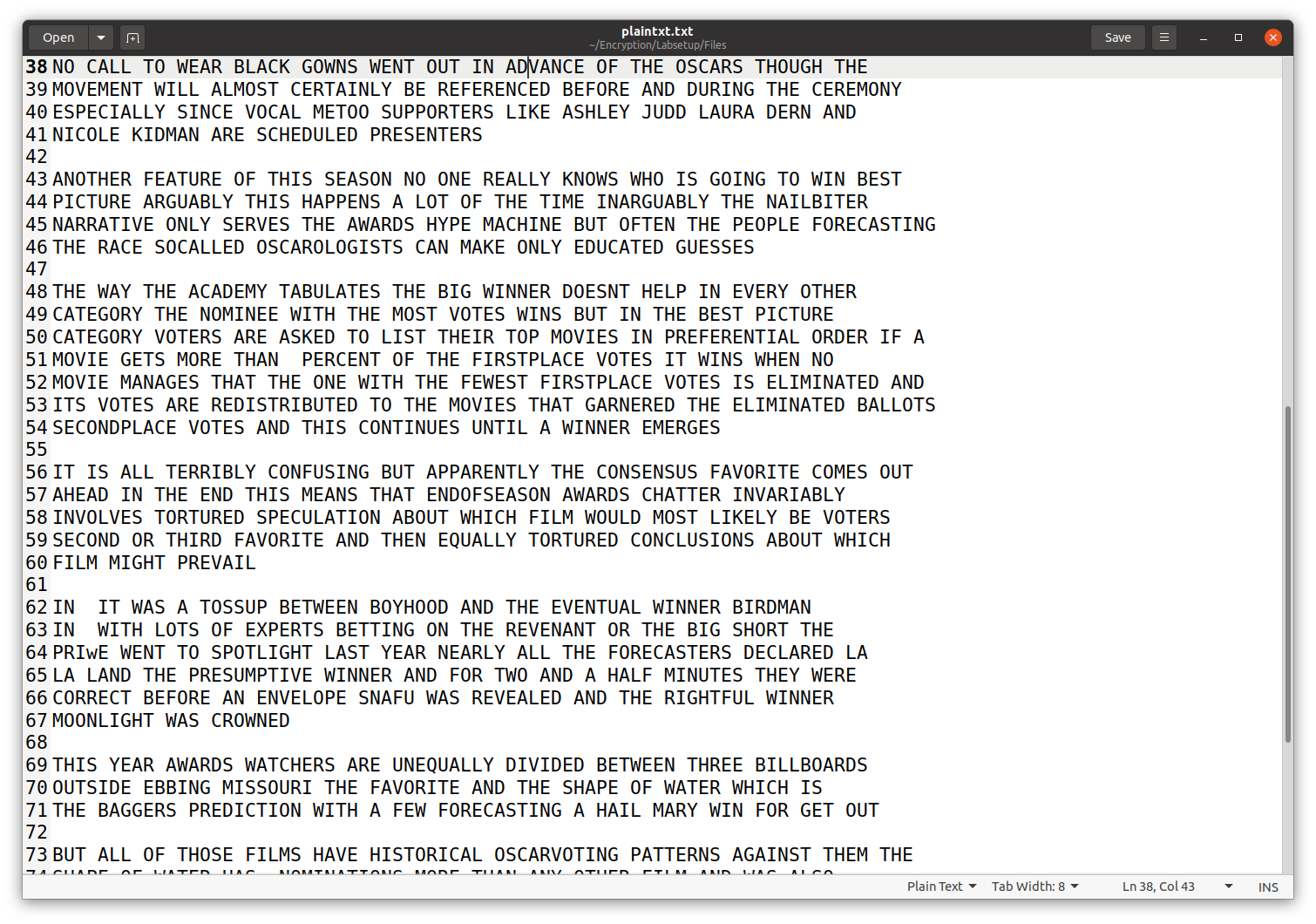
And then I deciphered the cipher text letter by letter as you can see from the screenshot.

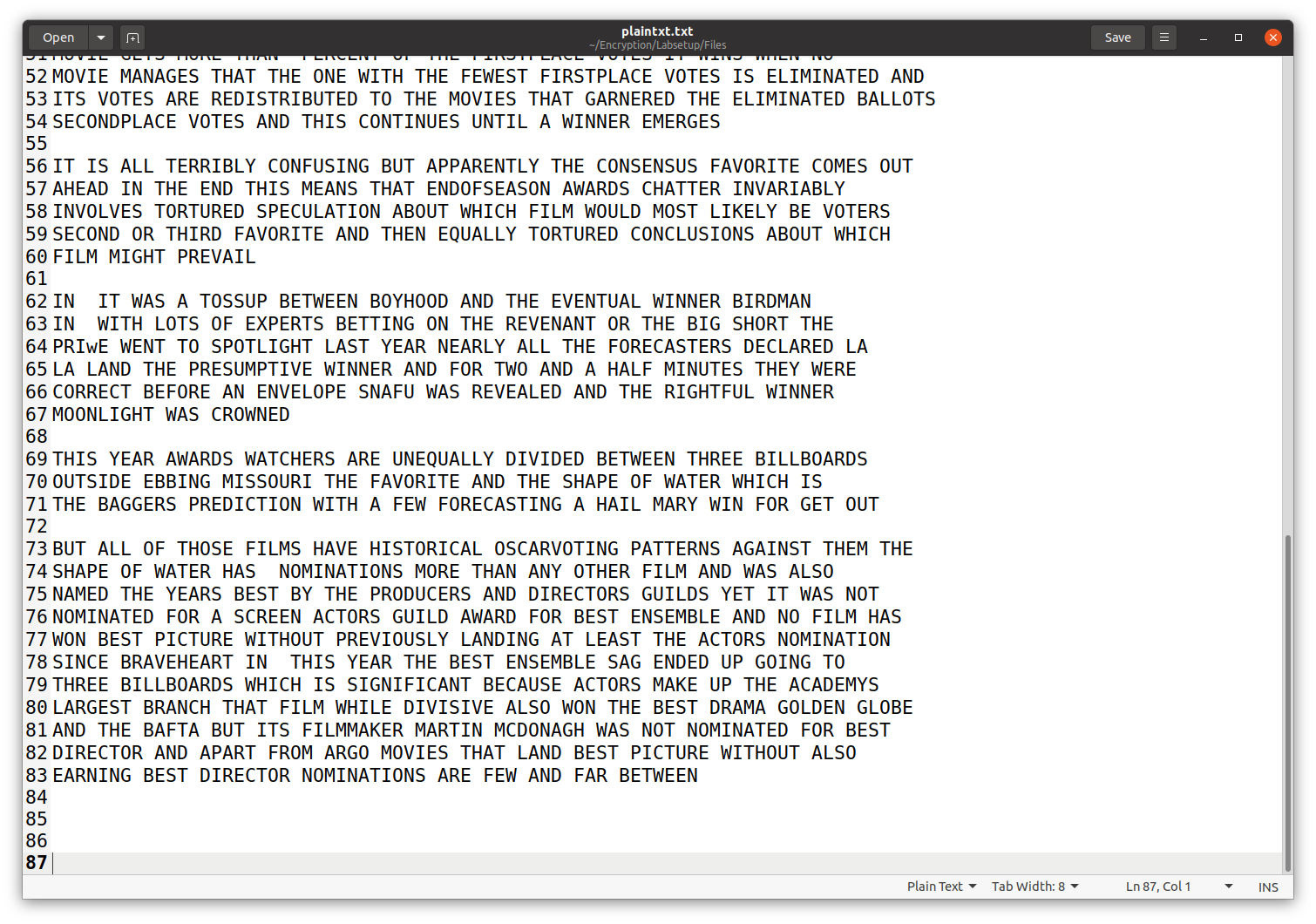




So this the cipher text.





So this is deciphered test you get as the final outcome.

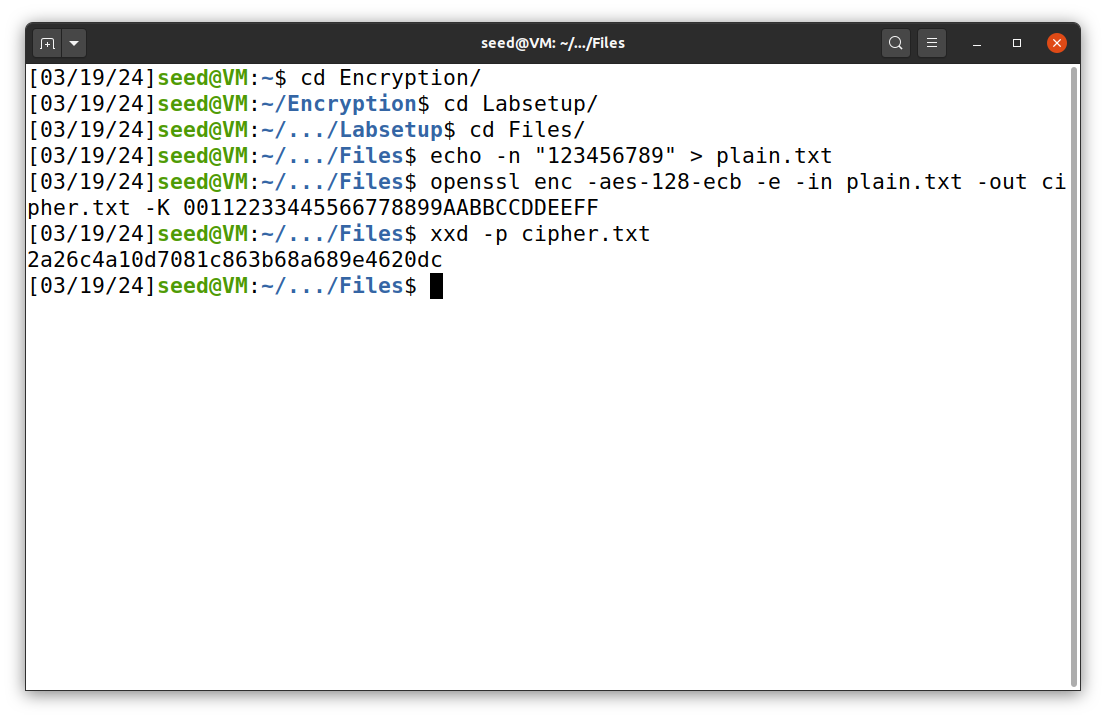
The encryption key that I found to encrypt this cipher is -

‘**ytnvupxlqamhzdrcgiesbfjok’ -> 'THEANDOWSCIRUYGMBLPKFVQJX'**

**Task 2: Encryption using Different Ciphers and Modes**

In this task, we experiment with 3 different kinds of encryption algorithms and modes.

I have used AES ECB, CBR and CTR algorithm modes.



This is **AES ECB** – This algorithm only needs a 16 byte key. Using xxd -p command we converted the text into hex string.

This is **AES CBC** – this mode requires an additional parameter of the initialization vector to be specified.

This is **AES CTR** – This mode also requires an additional parameter of initialization vector to be added.

Thus, we have successfully demonstrated three different modes of AES encryption algorithm and used it to encrypt plaintxt.

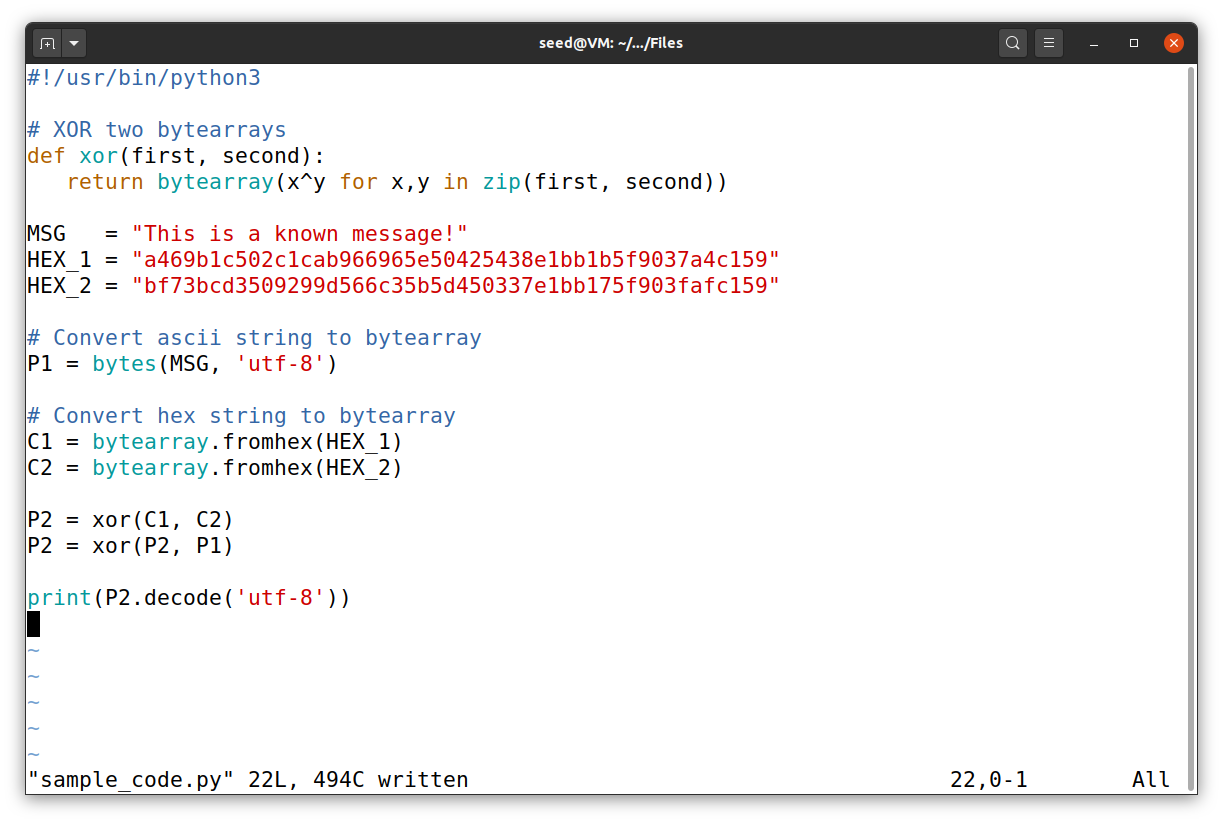
**Task 6: Initial Vector (IV) and common Mistakes**

**Task 6.1: IV Experiment**

In this subtask, we experiment with uniqueness of IV. We encrypt the plain.txt with same IV twice. This shows that the cipher text remains same when we use same IV. So this means no IV should be reused under same key.

When we try to encrypt them using different IV for the same plain text we get different output for both time. This shows that using different IV each time leads to different encyrption.

**Task 6.2: Common Mistake : Use the Same IV**

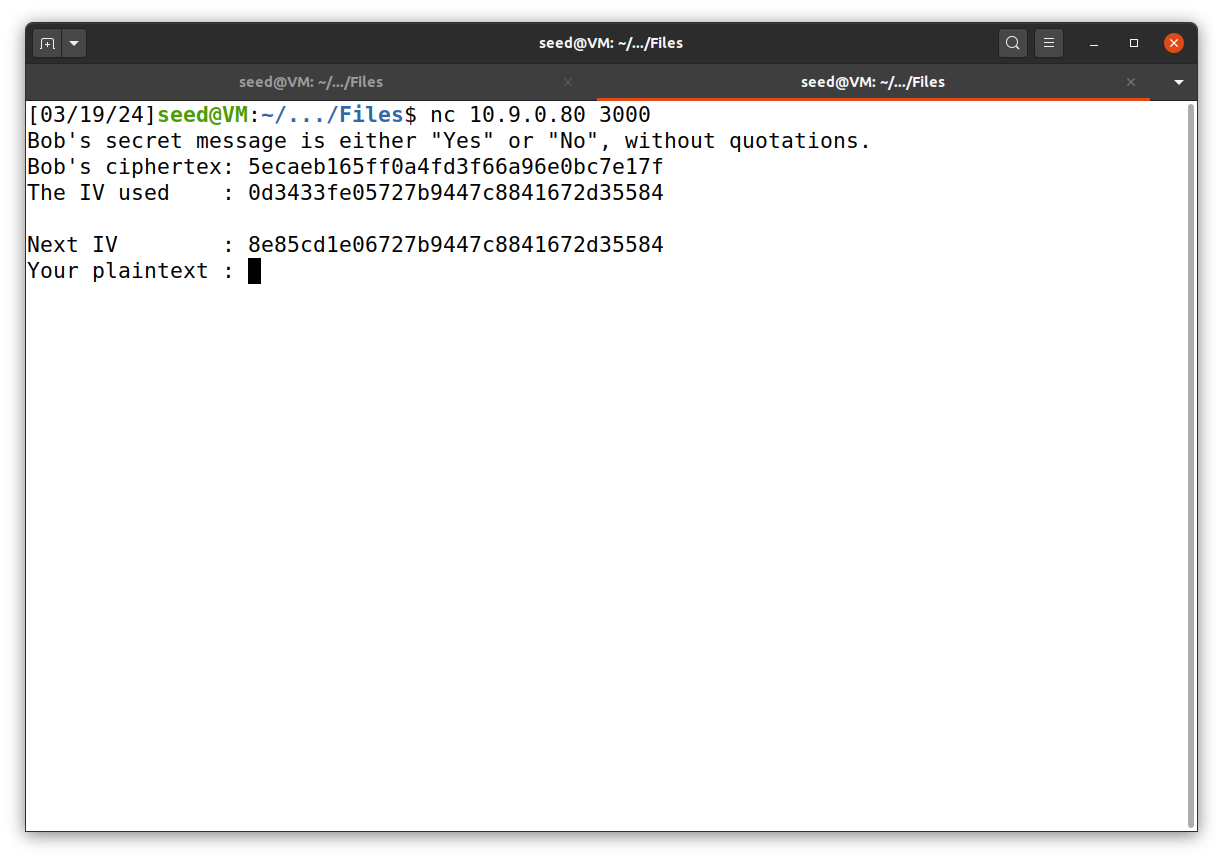
This is the code we will use for this sub task , as we are given P1, C1, and C2. Using xor we will cancel out the IV and get the answer for P2.

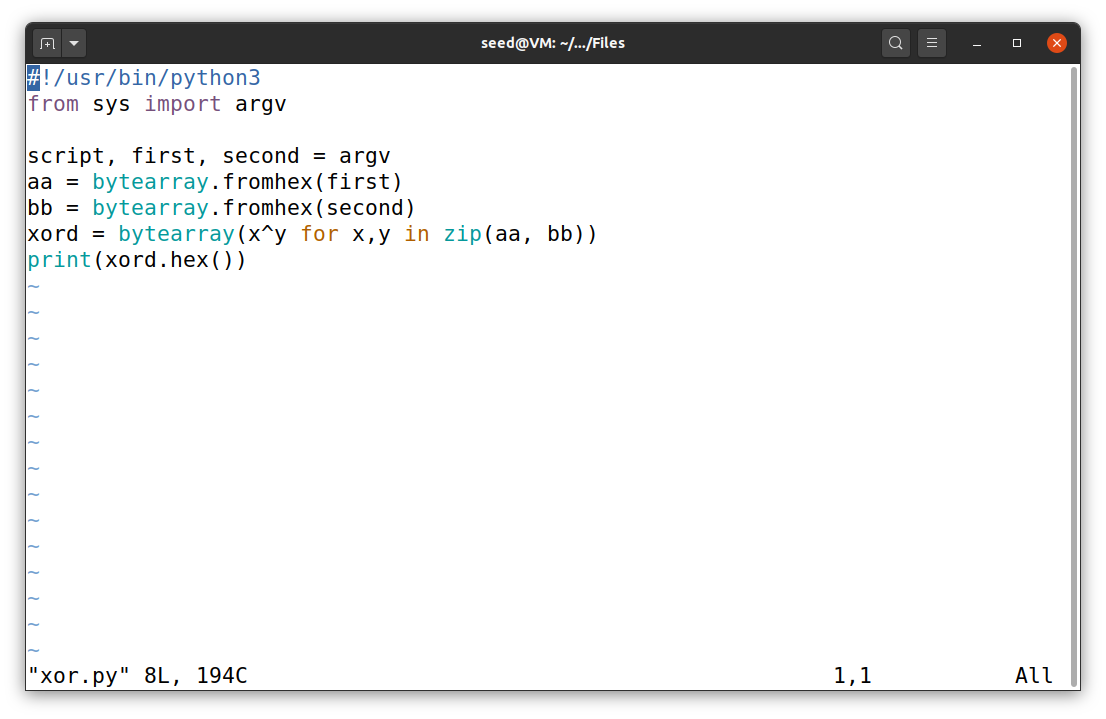
When we run the code this the output we get. Thus we get the P2 that is plain text – “Order: Launch a missile!”

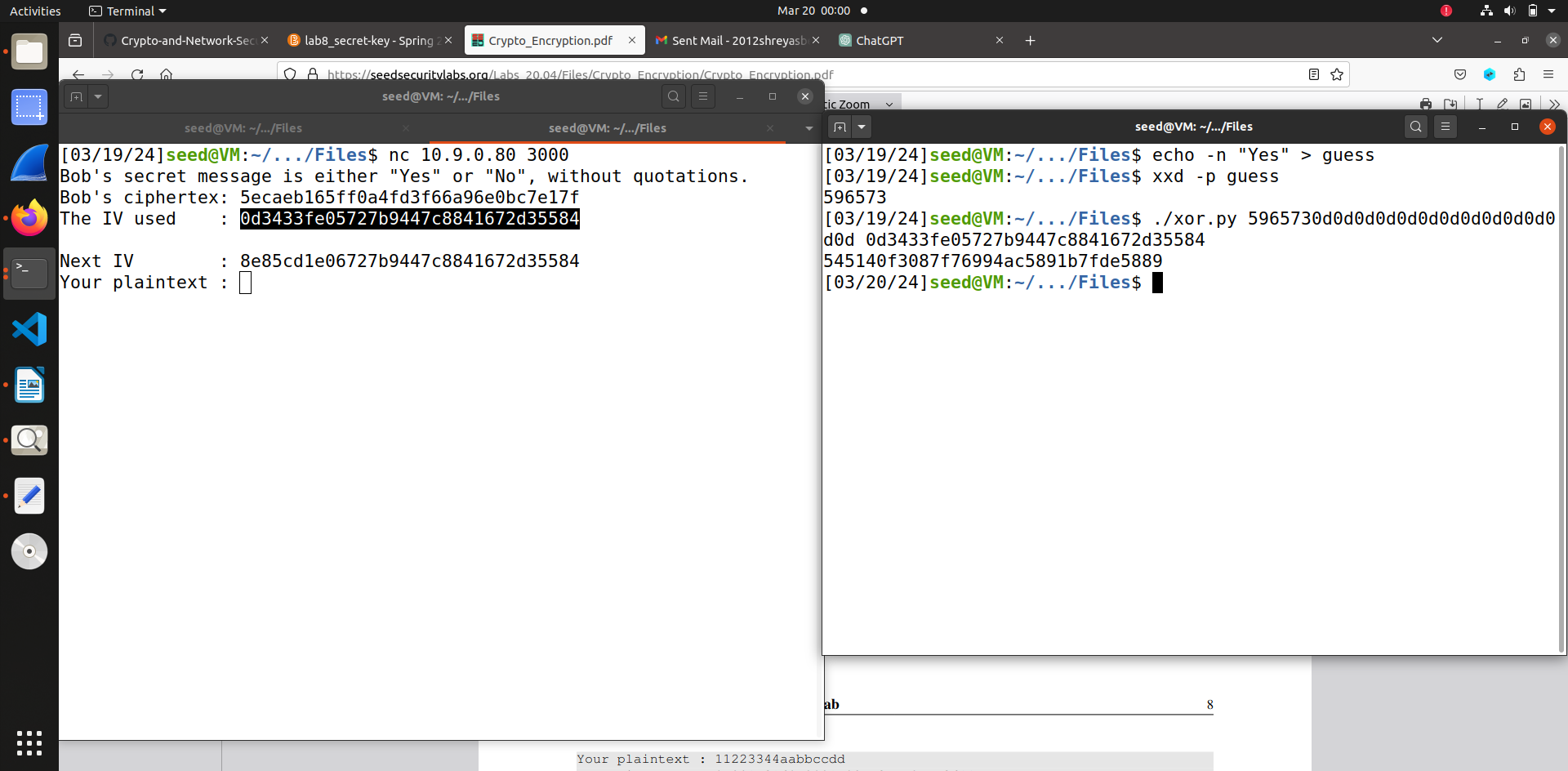
Replacing OFB with CFB -

If we replaced OFB with CFB, then the attacker would not be able to fully recover the plaintext.

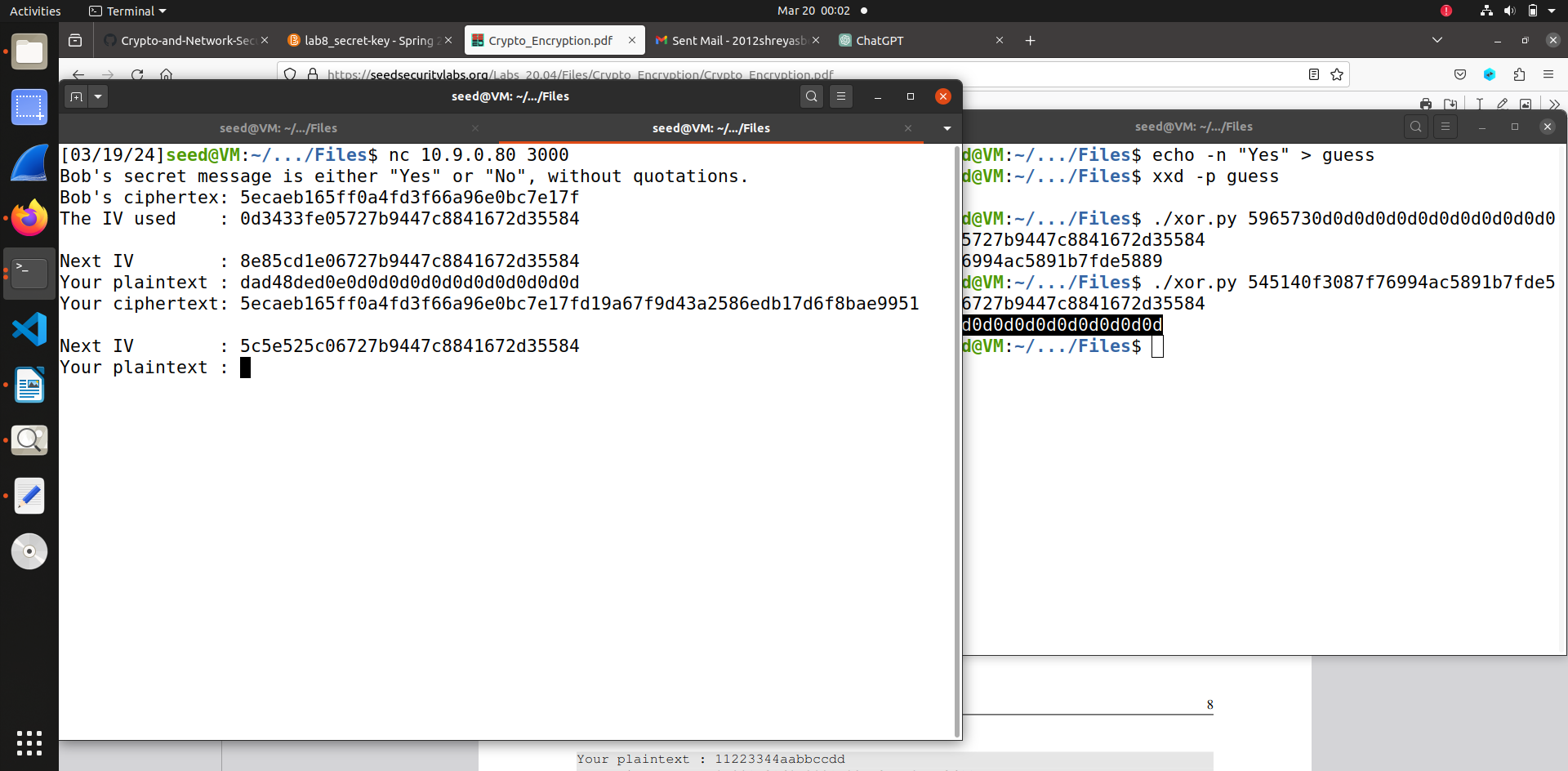
**Task 6.3: Common mistake : Use a Predictable IV**

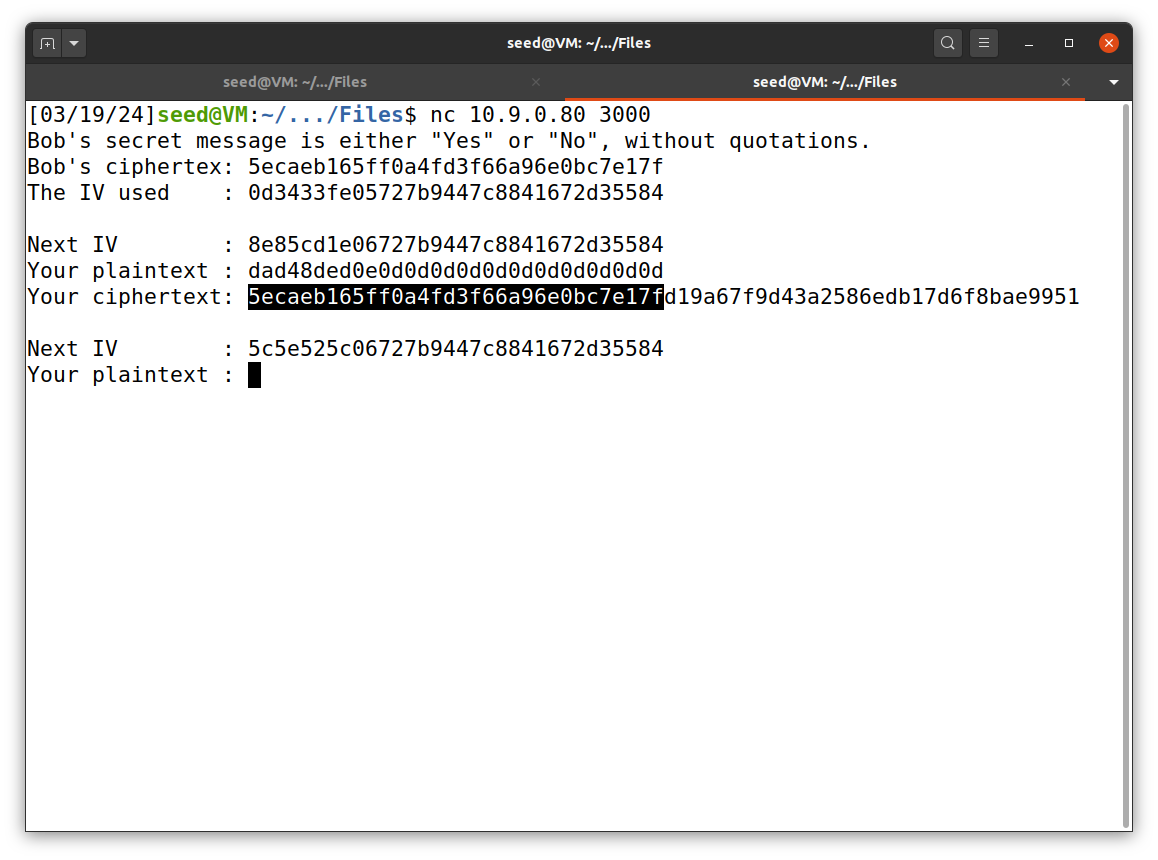
In this sub task, we need to figure out what Bob voted. SO first we connect to oracle, and run netcat command.

This is the xor code we will use for the hex strings.

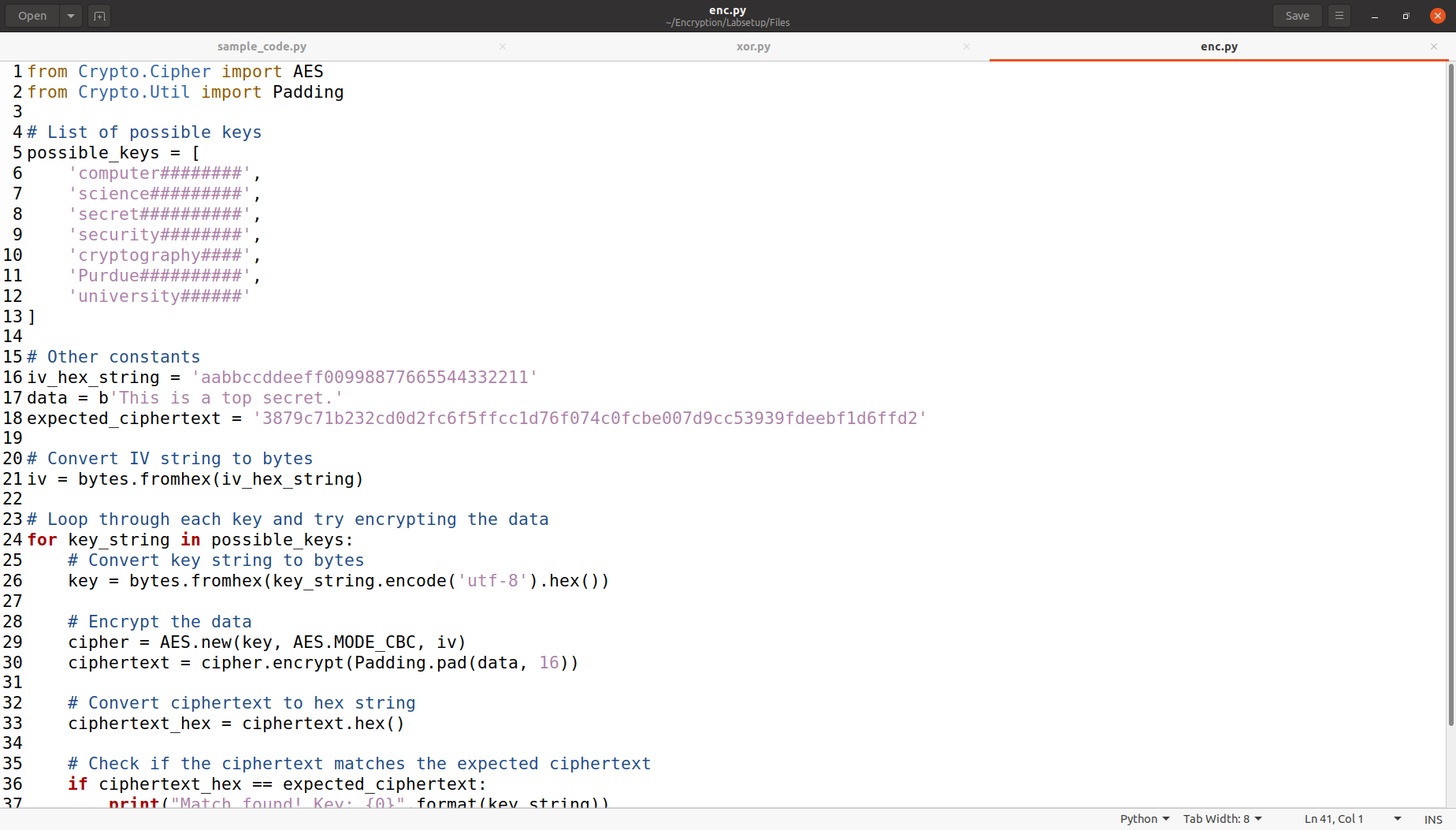
 Firstly, I guessed the vote as “Yes” and saved it in guess file, and then converted it into hex string using xxd -p command. The encryption mode used is CBC, the hex string 596573 has 3 bytes, so we need to add 13 more bytes for padding so we add 0d 13 times, and then xor with Bob’s IV.

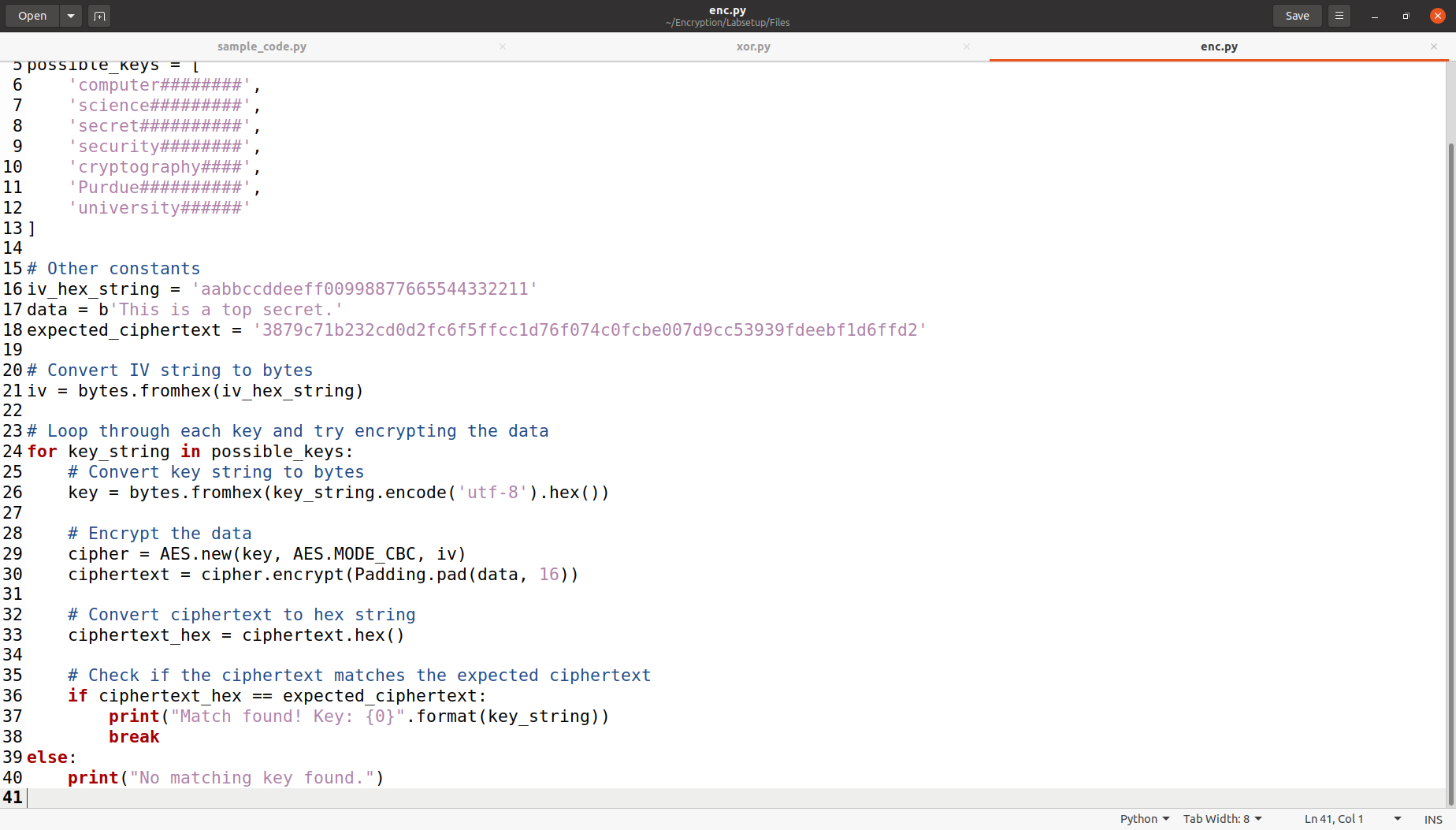
After which we get the new output.

After that we xor output we get previously with the next IV used to get our output. Which we then use as input for our plain text.

So as you can see from the screenshot our cipher text matches with Bob’s cipher text, which means we guessed the vote right which was “Yes”.

**Task 7: Programming using the Crypto Library.**



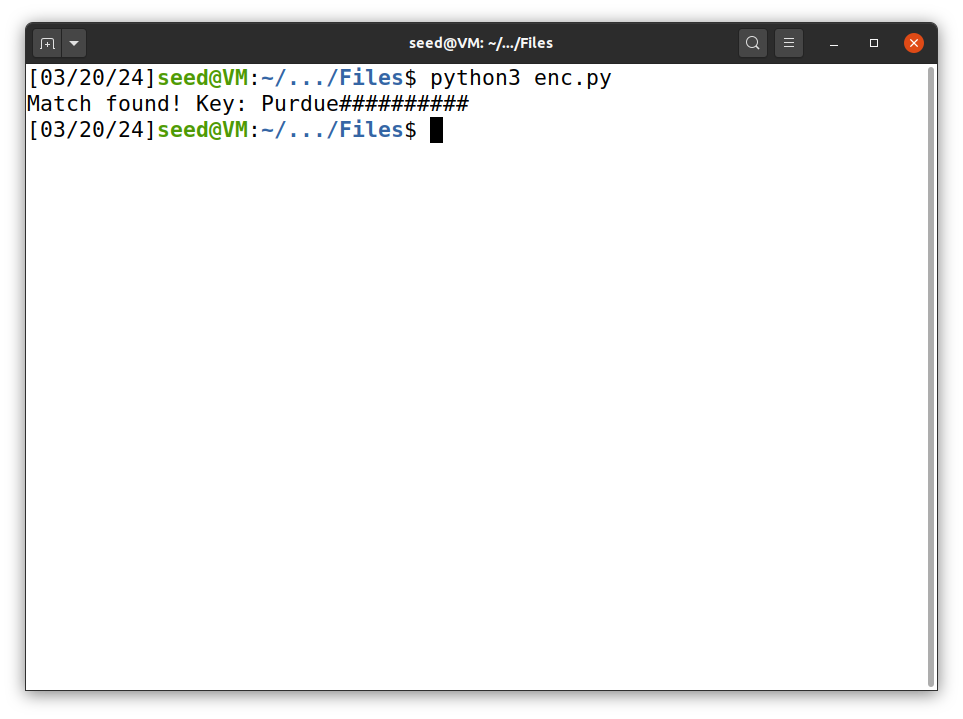
This is the code we will use , to find out the key used to encrypt a plain text from a list of encryption keys. If the obtained cipher text matches the expected cipher text, then we print the key used for encryption.

We are given the following -

The given plaintext : “This is top secret.”

Cipher text – 3879c71b232cd0d2fc6f5ffcc1d76f074c0fcbe007d9cc53939fdeebf1d6ffd2

IV in hex – aabbccddeeff00998877665544332211

Thus, we have successfully figured out the right encryption key used to encrypt the given plain text.