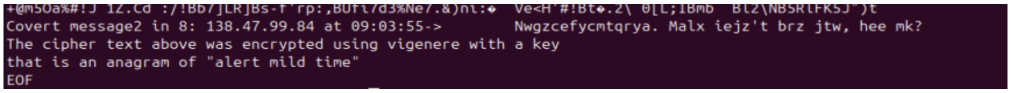
Fully Decoded Message:
138.47.99.84 at 09:03:55 ->   Congratulations. That wasn’t too bad, was it?
The cipher text above was encrypted using vigenere with a key that is an anagram of “alert mild time”

Disclaimer:
Our group didn’t fully get the message but a sample of some partially readable messages is in the text.txt file. With the help of Dr. Anky and his fancy computer, we were able to proceed through the challenge. Although, we were able to read some of the messages that had an IP address of the computer reading the during the chat program and a timestamp of when the program was run and then very broken words that were saying the cipher text above was encoded using vigenere …

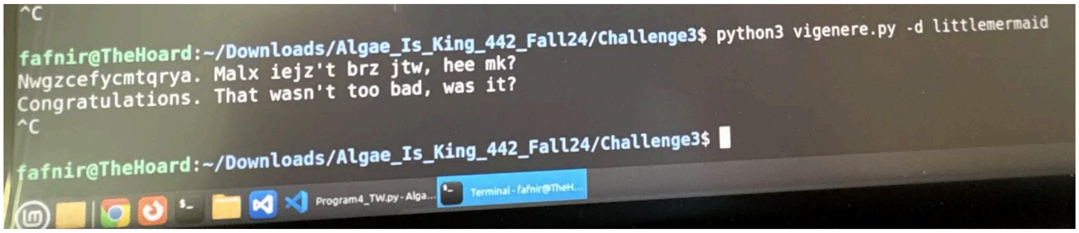
Steps to Complete Challenge:

1) First translate the beginningcipher.txt file to 0 and 1 using text_to_int.py and putting output into a new text file called binary.txt
2) Then taking the binary.txt to run through the binary_to_ascii.py to find message
3) Then there was another set of information that we have to decode using the vigenere.py with the key given “Prince Eric”
4) Found chat server: 138.47.99.160 port 12321 with running through vigenere.py 5 times.
5) Once we found the server we adjusted the program4TW.py file to listen from the given server.
6) We were not able to get the covert message
7) With the help of Dr. Anky running our Program4_TW_EDITED.py file on his computer, he was able to get a complete message. The image below shows the complete message.



```
+@m5Oa%#!J 1Z.Cd :/!Bb7]LR]Bs-f'rp:,BUfi7d3%Ne7.&)nl:◆  Ve<H'#!Bt◆.2\ 0[L;IBmb  Bl2\NB5RlFK5J")t
Covert message2 in 8: 138.47.99.84 at 09:03:55->        Nwgzcefycmtqrya. Malx iejz't brz jtw, hee mk?
The cipher text above was encrypted using vigenere with a key
that is an anagram of "alert mild time"
EOF
```

8) We then ran the “alert mild time” through an anagram converter and a got key of “littlemermaid”
9) We then ran the encrypted message through the vigenere decoder with the “littlemermaid” key to get the final message of “Congratulations. That wasn’t too bad, was it?” The image below shows us running the vigenere program to get the final output:



```
^C

fafnir@TheHoard:~/Downloads/Algae_Is_King_442_Fall24/Challenge3$ python3 vigenere.py -d littlemermaid
Nwgzcefycmtqrya. Malx iejz't brz jtw, hee mk?
Congratulations. That wasn't too bad, was it?
^C

fafnir@TheHoard:~/Downloads/Algae_Is_King_442_Fall24/Challenge3$ 
```

Different things we tried:

- In the program4_TW_EDITED.py file we edited it to print out a message if it was decoded in a 7 bit format or 8 bit format. We also edited it to where at a certain time will change it to 0 if it’s greater than or 1 if it’s less than and then opposite. So each time we run through the program to try and find the message it prints 4 different possibilities.

- We also ran through the program multiple times trying to change the time cutoff from 0.05, 0.06, 0.07, 0.08, 0.09.
- We realized the program actually ran better while not connected via ethernet and we would get partial parts of the message while just on wifi.

Contributions:

Kaylee: I helped run and try to get the covert message using program 4. I also tried many different ways to try to debug the program to try and get the correct output.

Andrew: I helped run Program 4 to try to get the covert message and then helped try to debug it because it was not working. I then used what we got from Dr. Anky to decode the final parts of the message using an anagram decodered and then running it through our vigenere program .

Gabe: I changed our python script that converts text to numbers to work with the beginning cipher given at the start of the challenge. I then put that binary output into a text file and ran it through our binary to ascii script. This gave me an encrypted message and a key which I plugged into our vigenere encoder/decoder script until it stopped outputting giberish. That output was an IP address and a port number which we then used to find the final covert message. Finally, I updated our vigenere script so that it will take a number as an argument in the command line and then encode or decode the given message that many times.

Hayden: While I helped with each step, I was generally behind the rest of the group. I did run point on debugging the timed message during class.

Tim: Used an online vigenere decoder on the encrypted message, but wasn't able to get the proper output. Tried to use FTP to get the message to print out, but my FTP/IP are messed up I think. Also ran the contents of newBinary through the binary_to_ascii.py program to try and find anything.

Tommy: Wrote the program used to connect to the server and decrypt the encoded message. Help with attempting to get the message through many trials and different methods.

John: Helped decode the beginning cipher file and then helped with trying to debug the program4 output.