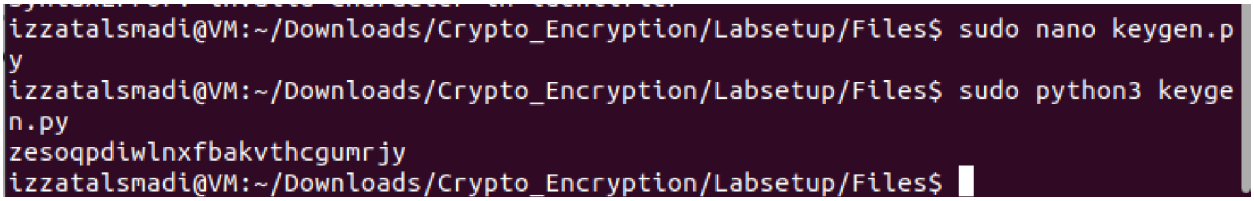
Quiz1

Secret Key Encryption Lab a sample walkthrough (without using the Docker container)

- **Open content file from SEEDLAB folder**

- **Start with Python code to generate key**

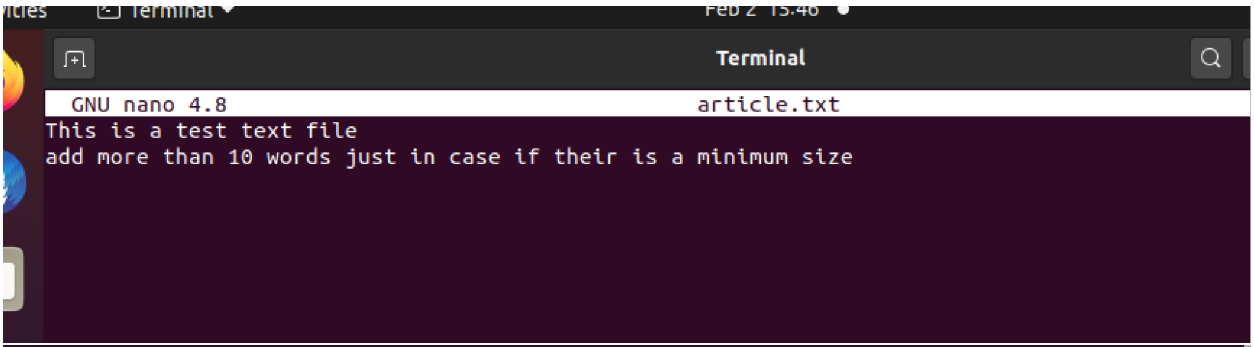
- **Compile and run**



- **Step 2: let us do some simplification to the original article. We convert all upper cases to lower cases, and then removed all the punctuations and numbers. We do keep the**

spaces between words, so you can still see the boundaries of the words in the ciphertext. In real encryption using monoalphabetic cipher, spaces will be removed. We keep the spaces to simplify the task. We did this using the following command

- **Create a demo test article.txt file**

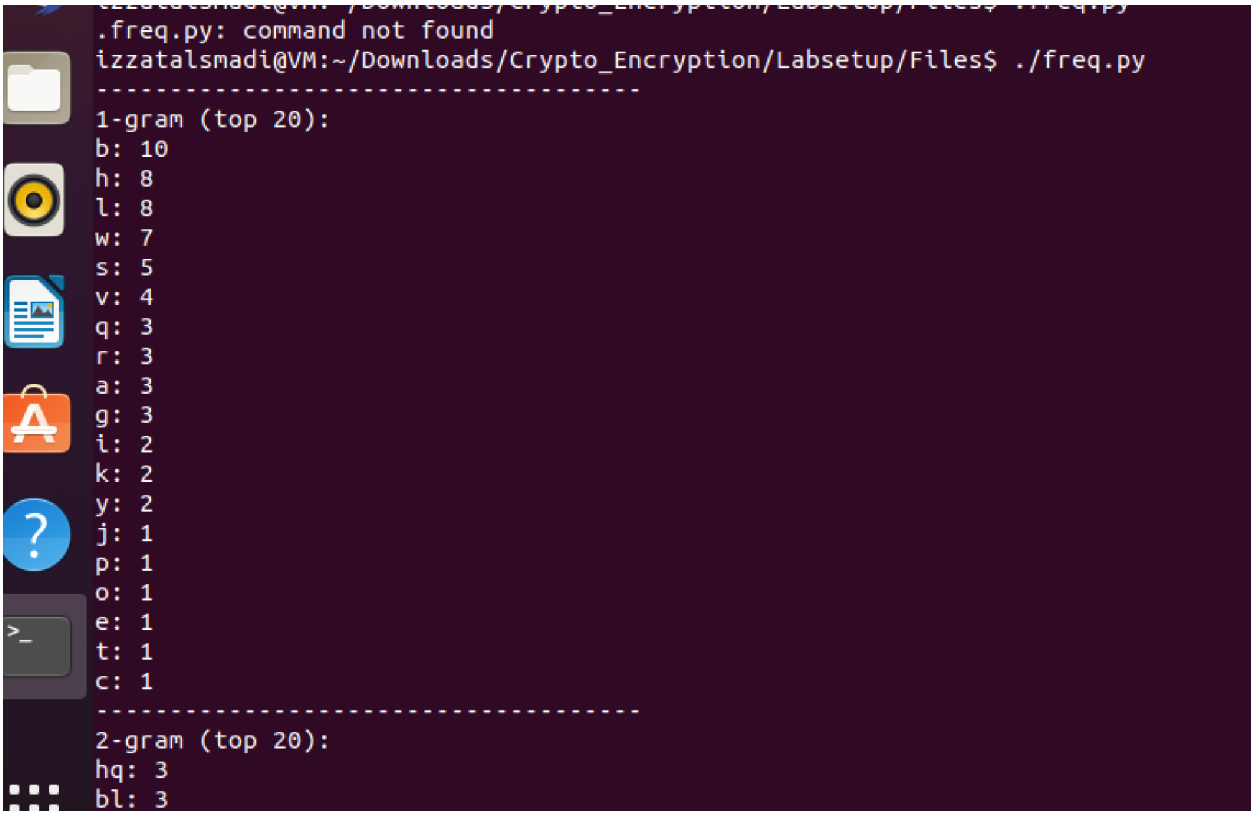



```
bash: article.txt: No such file or directory
izzatalsmadi@VM:~/Downloads/Crypto_Encryption/Labsetup/Files$ sudo nano article.txt
izzatalsmadi@VM:~/Downloads/Crypto_Encryption/Labsetup/Files$ tr [:upper:] [:lower:] < article.txt > lowerca
se.txt
izzatalsmadi@VM:~/Downloads/Crypto_Encryption/Labsetup/Files$ tr -cd '[a-z][\n][:space:]' < lowercase.txt >
plaintext.txt
izzatalsmadi@VM:~/Downloads/Crypto_Encryption/Labsetup/Files$
```

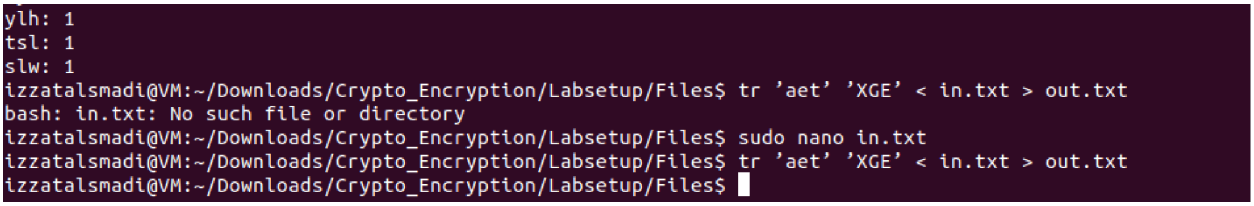
- **Step 3: we use thetrcommand to do the encryption. We only encrypt letters, while leaving the spaceand return characters alone.**

```
izzatalsmadi@VM:~/Downloads/Crypto_Encryption/Labsetup/Files$ tr -cd '[a-z][\n][:space:]' < lowercase.txt >
plaintext.txt
izzatalsmadi@VM:~/Downloads/Crypto_Encryption/Labsetup/Files$ tr 'abcdefghijklmnopqrstuvwxyz' 'sxtrwinqbedpv
gkfmalhyuojzc' \< plaintext.txt > ciphertext.txt
tr: extra operand '<'
Try 'tr --help' for more information.
izzatalsmadi@VM:~/Downloads/Crypto_Encryption/Labsetup/Files$ tr 'abcdefghijklmnopqrstuvwxyz' 'sxtrwinqbedpv
gkfmalhyuojzc' < plaintext.txt > ciphertext.txt
izzatalsmadi@VM:~/Downloads/Crypto_Encryption/Labsetup/Files$
```

- **We have created a ciphertext using a different encryption key (not the one described above). It is included in Labsetup.zip file, which can be downloaded from the lab's website. Your job is to use the frequency analysis to figure out the encryption key and the original plaintext. We have also provided a Python program (freq.py) inside the Labsetup/Files folder. It reads the ciphertext.txt file, and produces the statistics for n-grams, including the single-letter frequencies, bigram frequencies (2-letter sequence), and trigram frequencies (3-letter sequence), etc.**

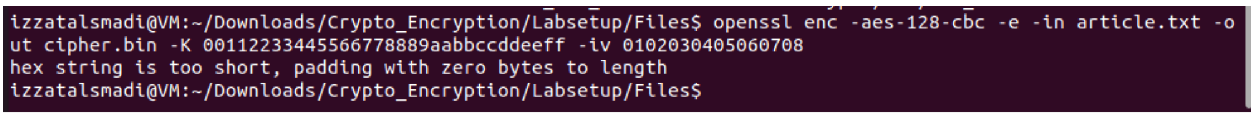


- **For example, in the following, we replace letters a,e, and t in in.txt with letters X,G,E, respectively; the results are saved inout.txt.**

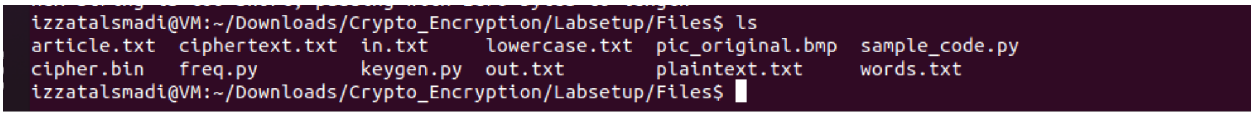


Task 2: Encryption using Different Ciphers and Modes

- **In this task, we will play with various encryption algorithms and modes. You can use the following openssl enc command to encrypt/decrypt a file. To see the manuals, you can type man openssl and man enc.**



- **Check the new file**



Task 3: Encryption Mode – ECB vs. CBC

- **Continue to the rest of the tasks**