- **Makefile commands**
    - default:
        - runs a frequency analysis on the given cipher text and outputs the occurrences of each letter and the 30 most common bigrams and 30 most common trigrams
    - run:
        - runs a frequency analysis on some un encoded text to test the accuracy of the program, the first text is chapter 4 of alice in wonderland and the output is saved as 'aliceCh4Output.txt'
        - the second text is chapter 8 of Alice in wonderland and the output is saved as 'aliceCh8Output.txt'
    - clean:
        - removes the output text files for each make command
    - decrypt:
        - runs a frequency analysis on some encoded text and generates a list of possible matching letters for each letter in the encoded text based on their frequency in the text compared to their actual frequency in the English language.
    - translate:
        - Using a hard coded translation alphabet, replaces each letter in an encoded text with their matching translation in the alphabet, producing a decoded text.

a) Run the program with the name of the text file to be analyzed as a command line argument

   a. Python3 HW1code_Brady_Messer.py inputText.txt

      i. Or run 'make'

   b. Program behavior is as follows:

      i. The string "xxxx" will be counted as 4 occurrences of x, 3 bigrams, and 2 trigrams

b) Output for Alice in wonderland chapter 4:

```
Overall character occurences:
e: 1265
t: 1054
a: 878
o: 833
h: 733
n: 705
i: 702
s: 607
r: 537
l: 486
```

```
d: 474
u: 303
g: 262
w: 261
m: 215
f: 212
c: 207
b: 183
y: 177
p: 161
k: 100
v: 64
x: 16
q: 14
j: 7
z: 5
--------------------
Bigram occurences:
('ee', 118)
('et', 98)
('ae', 92)
('te', 89)
('tt', 80)
('eo', 80)
('ot', 76)
('he', 73)
('at', 70)
('ei', 69)
('ea', 69)
('oe', 68)
('to', 66)
('ta', 65)
('ne', 65)
('es', 64)
('eh', 63)
('th', 62)
('ie', 61)
('aa', 58)
('en', 58)
('er', 56)
('tn', 55)
('el', 55)
('it', 54)
('ht', 54)
```

```
('ti', 54)
('re', 53)
('ed', 52)
('ia', 51)
--------------------
Trigram occurences:
('the', 40)
('ehe', 34)
('ihe', 26)
('eth', 25)
('ath', 23)
('ohe', 20)
('aan', 19)
('ith', 19)
('nhe', 18)
('ean', 18)
('ahe', 18)
('eou', 18)
('tth', 18)
('end', 17)
('eer', 17)
('lth', 17)
('oth', 17)
('hhe', 17)
('tan', 16)
('ein', 16)
('hth', 15)
('dhe', 15)
('tit', 15)
('nth', 15)
('rhe', 15)
('lhe', 14)
('eto', 14)
('sth', 14)
('nin', 13)
('dth', 13)
```

Output for Alice in wonderland chapter 8:

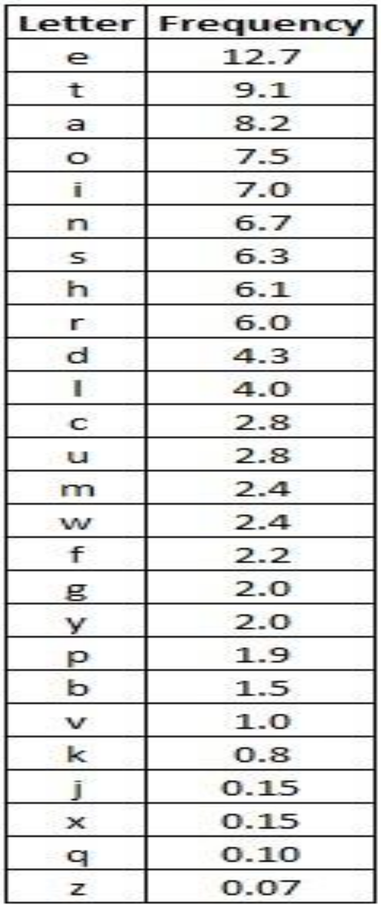
```
Overall character occurences:
e: 1453
t: 965
a: 840
o: 736
```

```
h: 734
n: 718
i: 670
s: 615
r: 515
d: 500
l: 418
u: 329
g: 265
w: 256
c: 240
f: 200
y: 193
m: 181
p: 127
b: 123
k: 103
v: 90
q: 47
x: 24
j: 13
z: 4
--------------------
Bigram occurences:
('ee', 167)
('te', 110)
('et', 105)
('eo', 87)
('en', 87)
('ae', 82)
('ea', 79)
('ne', 79)
('eh', 77)
('oe', 76)
('ie', 73)
('he', 69)
('ei', 66)
('tt', 65)
('ta', 64)
('se', 62)
('aa', 59)
('es', 59)
('tn', 59)
('at', 59)
```

```
('le', 56)
('ot', 54)
('re', 53)
('er', 52)
('na', 51)
('oo', 50)
('de', 50)
('oa', 49)
('it', 49)
('ha', 49)
--------------------
Trigram occurences:
('ehe', 50)
('eth', 44)
('the', 33)
('tth', 27)
('nhe', 25)
('ath', 23)
('hhe', 23)
('ein', 21)
('ith', 21)
('oth', 20)
('ean', 20)
('ohe', 20)
('rhe', 19)
('eou', 19)
('eer', 18)
('nth', 17)
('rth', 17)
('end', 17)
('tin', 17)
('eed', 16)
('ihe', 15)
('ner', 15)
('hin', 15)
('ahe', 15)
('eng', 15)
('hth', 15)
('aer', 15)
('lth', 14)
('een', 14)
('dhe', 14)
```

c) The two sets of characters are very similar, the first 9 most common characters are exactly the same for each analyzed chapter of Alice in wonderland, but the frequencies of each character are slightly different. The most common bigram for each chapter was 'ee' and the most frequent trigrams for each chapter are similar but slightly different.

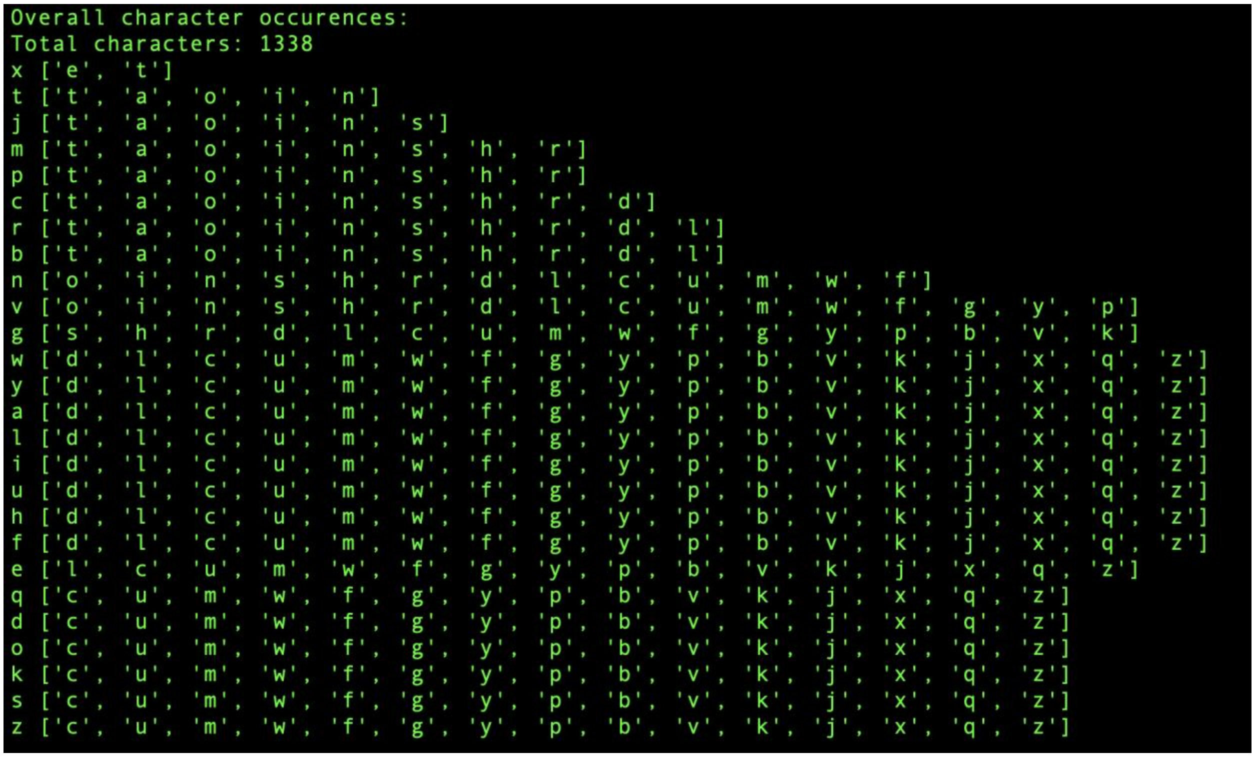
d) According to https://crypto.interactive-maths.com/frequency-analysis-breaking-the-code.html the frequencies of each letter of the English alphabet is as follows:



Letter	Frequency
e	12.7
t	9.1
a	8.2
o	7.5
i	7.0
n	6.7
s	6.3
h	6.1
r	6.0
d	4.3
l	4.0
c	2.8
u	2.8
m	2.4
w	2.4
f	2.2
g	2.0
y	2.0
p	1.9
b	1.5
v	1.0
k	0.8
j	0.15
x	0.15
q	0.10
z	0.07

a. Using these frequencies, I made a program called decrypt.py which takes in a given encoded text, performs a frequency analysis of the text, creates a list of possible alphabets with each possible matching letter being within +- 3% frequency to account for the smaller sample size for the encoded text. Then it goes through the list of possible matches and randomly generates alphabets for the encoded text, and using these alphabets it replaces every letter in the encoded text with what could be its decoded counterpart. Using this plus some intuition gives me the following conclusions:

i. Here is the list of possible matches for each encoded letter given by my decryption program:



```
Overall character occurences:
Total characters: 1338
x ['e', 't']
t ['t', 'a', 'o', 'i', 'n']
j ['t', 'a', 'o', 'i', 'n', 's']
m ['t', 'a', 'o', 'i', 'n', 's', 'h', 'r']
p ['t', 'a', 'o', 'i', 'n', 's', 'h', 'r']
c ['t', 'a', 'o', 'i', 'n', 's', 'h', 'r', 'd']
r ['t', 'a', 'o', 'i', 'n', 's', 'h', 'r', 'd', 'l']
b ['t', 'a', 'o', 'i', 'n', 's', 'h', 'r', 'd', 'l']
n ['o', 'i', 'n', 's', 'h', 'r', 'd', 'l', 'c', 'u', 'm', 'w', 'f']
v ['o', 'i', 'n', 's', 'h', 'r', 'd', 'l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p']
g ['s', 'h', 'r', 'd', 'l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k']
w ['d', 'l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
y ['d', 'l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
a ['d', 'l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
l ['d', 'l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
i ['d', 'l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
u ['d', 'l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
h ['d', 'l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
f ['d', 'l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
e ['l', 'c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
q ['c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
d ['c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
o ['c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
k ['c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
s ['c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
z ['c', 'u', 'm', 'w', 'f', 'g', 'y', 'p', 'b', 'v', 'k', 'j', 'x', 'q', 'z']
```

ii. The trigram ‘jpx’ occurs very frequently in the text and the word that makes the most sense to map to this is ‘the’

iii. The trigram ‘mtn’ also occurs frequently, and I deduced this one stands for ‘and’ since ‘mtn’ occurs inside of longer words and ‘jpx’ doesn’t occur inside of other words as often.

iv. The bigram ‘bt’ occurs very often and the word that makes the most sense to map this to is ‘in’

v. My final alphabet mapping is the following (after a lot of trial and error)

Encoded letter	Decoded letter
x	e
t	n
j	t
m	a
p	h
c	o
r	s
b	i
n	d
v	r
g	l
w	g
y	w
a	c
l	m
i	f

u	u
h	k
f	b
e	y
q	p
d	v
o	z
k	q
s	j
z	x

vi. Using this mapping, I wrote a program called translate.py which takes a specific character mapping and replaces each letter in the cipher text with its counterpart, running ‘make translate’ will run this program and produce a deciphered output. My deciphering of the text is the following:

in the same hour came forth fingers of a man's hand, and wrote over against the candlestick upon the plaster of the wall of the king's palace; and the king saw the part of the hand that wrote. then the king's countenance was changed, and his thoughts troubled him, so that the joints of his loins were loosed, and his knees smote one against another. the king cried aloud to bring in the astrologers, the chaldeans, and the soothsayers. and the king spake, and said to the wise men of babylon, whosoever shall read this writing, and show me the interpretation thereof, shall be

clothed with scarlet, and have a chain of gold about his neck, and shall be the third ruler in the kingdom. then came in all the king's wise men; but they could not read the writing, nor make known to the king the interpretation thereof. then was king belshazzar greatly troubled, and his countenance was changed in him, and his lords were astonished. now the queen, by reason of the words of the king and his lords, came into the banquet house; and the queen spake and said, o king, live forever; let not thy thoughts trouble thee, nor let thy countenance be changed; there is a man in thy kingdom, in whom is the spirit of the holy gods; and in the days of thy wisdom of the gods, was found in him; whom the king nebuchadnezzar thy father, the king, i say, thy father, made master of the magicians, astrologers, chaldeans, and soothsayers; forasmuch as an excellent spirit, and knowledge, and understanding, interpreting of dreams, and showing of hard sentences, and dissolving of doubts, were found in the same daniel, whom the king named belteshazzar; now let daniel be called, and he will show the interpretation.