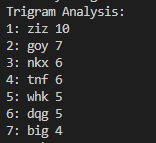
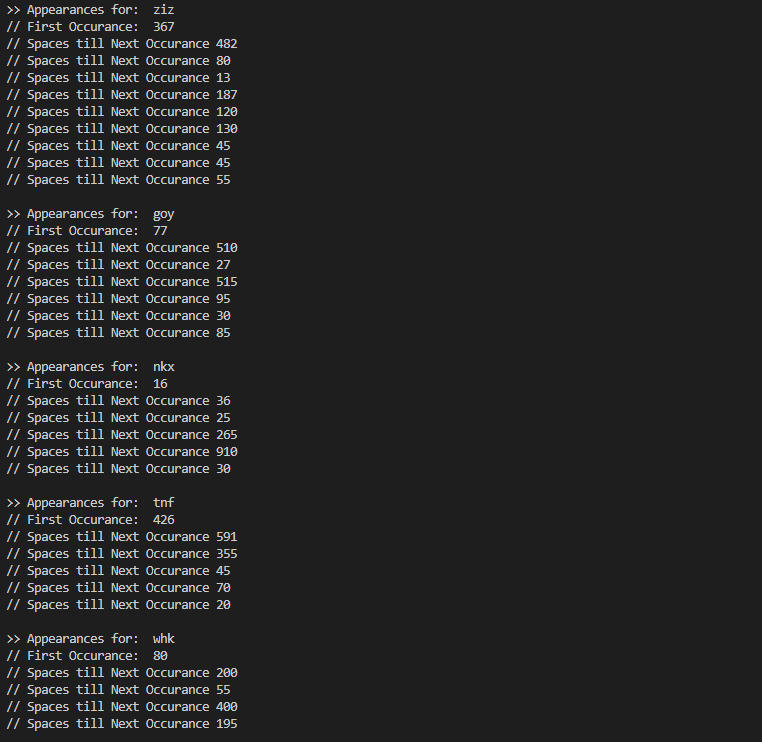
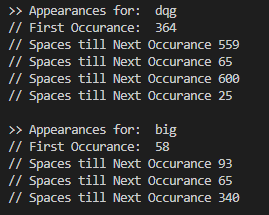
1. **List the five most common trigrams:**
2. Listing seven, as we have a few ties, here are our top five common trigrams:



1. **Show the difference between the starting indexes of the five most common trigrams:** 
   1. Same as before, listing seven – here are the occurrences:

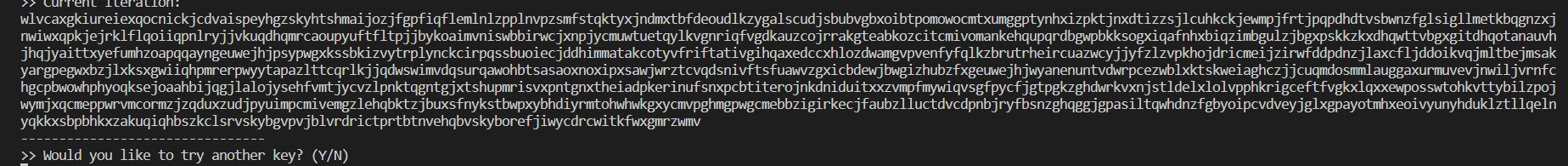


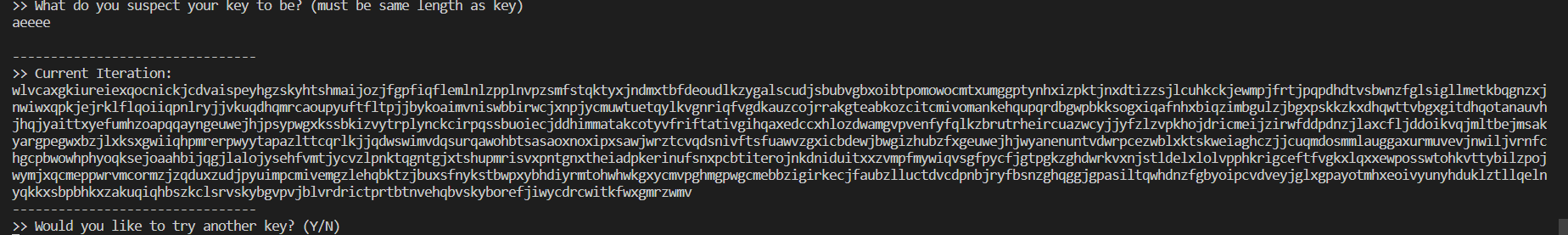
1. **Based on our findings, what do you suspect the key length is?** 
   1. Based on what we could see from the differences from the indexes of the occurrence, I suspect the key length to be 5, either a word or some random characters.
   2. Specifically, if we look at some of the occurrences, while we progress through the occurrences, we begin to see multiples of 5 as seen above.
2. **Separate the ciphertext into X shift-by-N ciphers where X is the length of the key and perform monoalphabetic frequency analysis on each. What are the three most common ciphertext characters in each of the shift-by-N ciphers?** 
   1. For starters, when running my Python code – I allowed the user to pick which character ‘i’, they would like to shift (with 1 being the first character in the potential X, and X being the maximum character in that sequence a user can shift). This can be tested after the completion of the frequency analysis in the program.
   2. As seen here, we selected a key that is of length 5, here is our individual frequency analysis for each shift-by-N cipher (next page):

Text

Description automatically generated

1. **Iterations until Key is found:**

1st Attempt, we shifted every character by E.

2nd – 6th Attempt, since we do not know if the key is a word or a set of characters – we go with a safe route and since we know that the most common starting letters for the alphabet are t,a,o,d,w – we try all of them. We come to find that this doesn’t really clarify much:

7th – 14th attempt: with not much discernible words, we attempt (in order) the most frequent letters for the second frequency paired with our previously stated set of five letters that we started with for the first character (keep in mind, the rest of the letter sequence is still shifted by e).

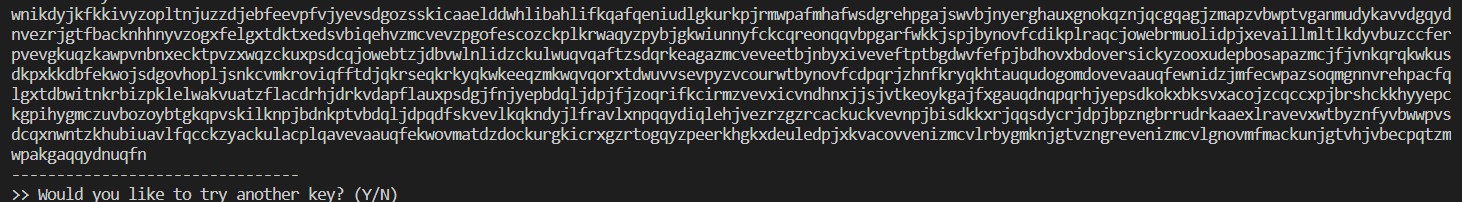
A screenshot of a computer

Description automatically generated with medium confidence

As seen above, we have yet to find a key – now with a string of characters to choose from for both the first and second characters – we begin to pick the potential 3rd character by working our way down the list of appearances. Ideally, we luck out here – as this has essentially become a brute forcing of this cipher.

15th – 23rd attempt.

At this point, I realized, that I my math was a bit off in my tool. See, rather than subtracting the shift amount – and cycling through – I was adding it. Resulting in a never jumble of letters. I tried the experiment all over again. Beginning with everything shifted by E.



Using the same strategy we employed earlier, we then move onto a combination for the first and second character in the sequence that potentially works.

A picture containing text

Description automatically generated

Using our intuition (this is probably around attempt 10 in the new go), we see a few words that could be formed by shifting the third character by 2.

Timeline

Description automatically generated with medium confidence

Following the same strategy, we move onto the fourth characters in this cipher:

A picture containing timeline

Description automatically generated

And finally, we move onto the final character:

A screenshot of a computer

Description automatically generated with medium confidence

Finally, we find that our final key is “AGBVD,” after around 20 attempts.

A screenshot of a computer

Description automatically generated with medium confidence