**Approaches**

***Vigenère Cipher***

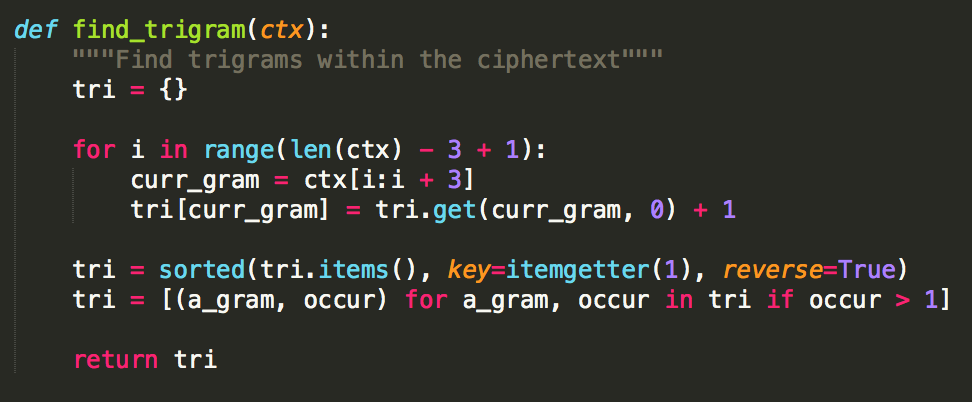
For the Vigenère cipher, cryptanalysis is broken down using the following primary functions:

**vigenere** returns the key and plaintext of a ciphertext encrypted via the Vigenère cipher

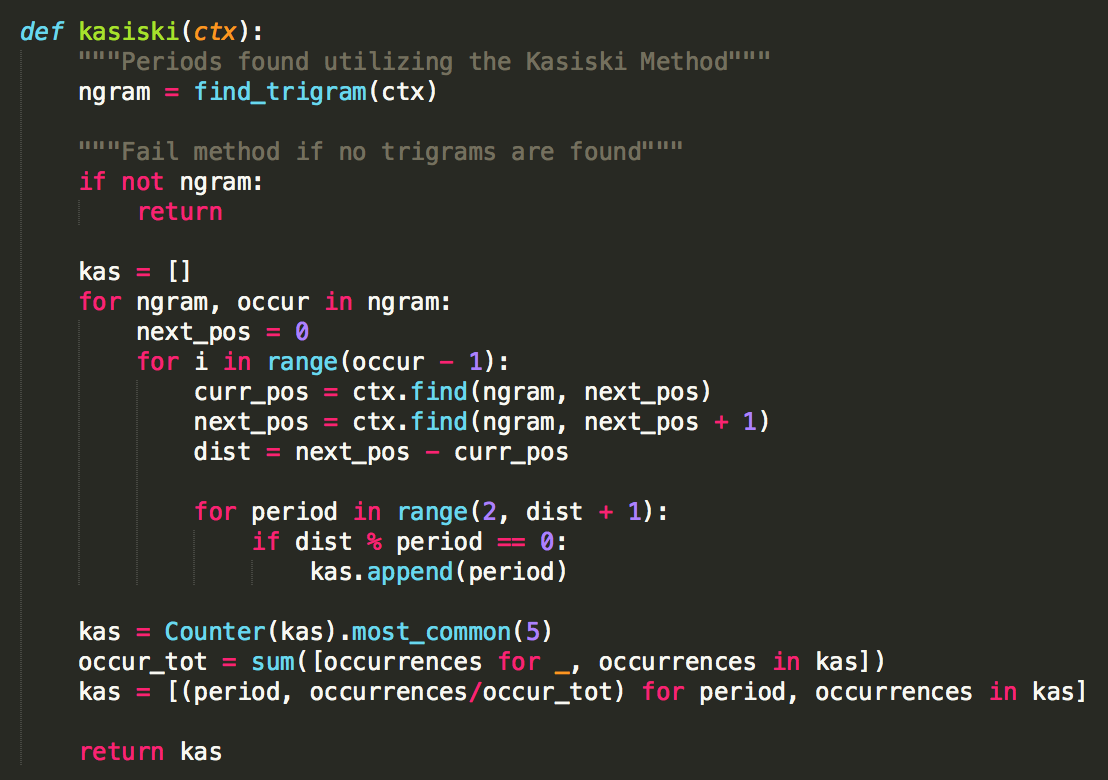
**compute\_periods\_with\_ic** returns the period of a polyalphabetic cipher

**Polyalphabetic cipher period**

Since Vigenère cipher uses a form of polyalphabetic substitution, the period needs to be found and analyzed as part of the cryptanalysis. The period is the number of letters encrypted before a polyalphabetic substitution cipher returns to its first cipher alphabet. And to obtain said period, the first step to take is to utilize the Kasiski method. Using the function **kasiski**, its first step is to step into another function, **find\_trigram**, to find the any existing trigrams that are repeated within the ciphertext.

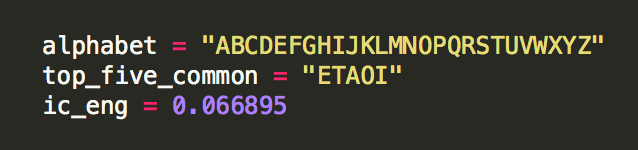


After finding all of the existing trigrams, the distance between the repeating trigrams is calculated. After calculating the factors, the 5 most frequent are considered as possible periods of the ciphertext, concluding the Kasiski method function.

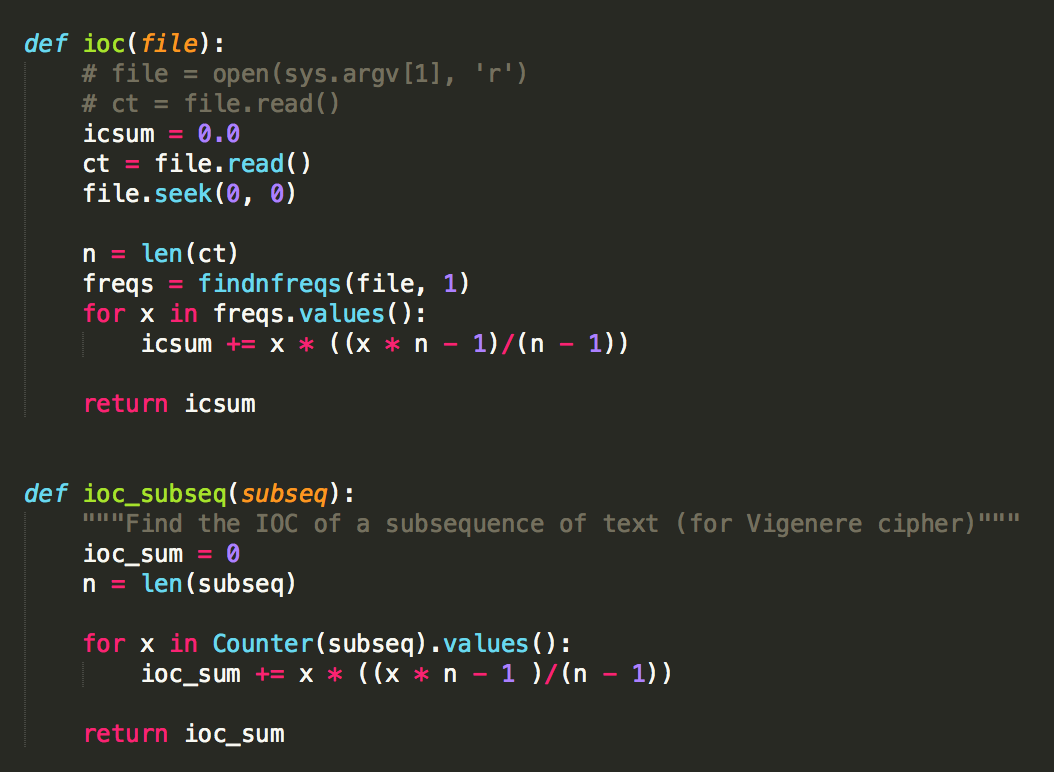


To automate the process of running the Kasiski method, it was decided that choosing possible periods based off of those that appear more often as factors in the distance of trigrams was the best route. Typically, the Kasiski method a GCD is made of the distances between trigrams to find an estimated period. This leaves a lot to chance due to potential repetition of the trigram, causing a GCD of 1, making it suboptimal for our use case. What is returned from this function is a list of the top 5 periods alongside their probabilities of being real periods.

For this cipher, a few pre-defined variables are necessary. In this case, the index of coincidence (IOC) of the English language, English alphabet, and top five common letter in the English language are required.



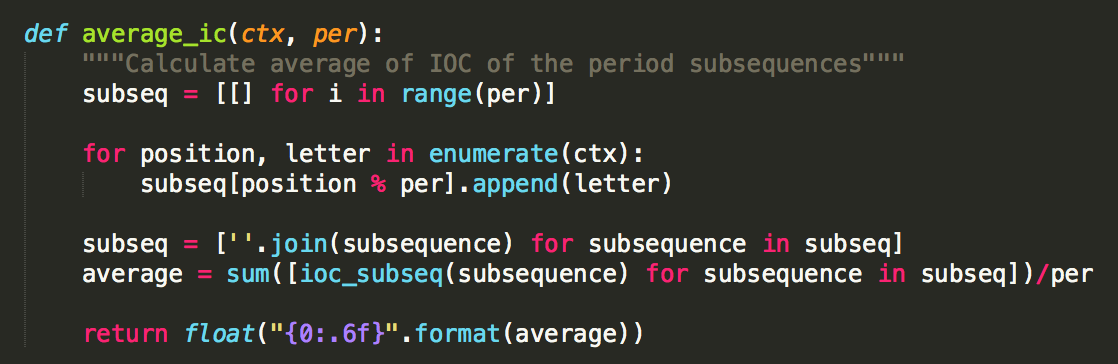
The English IOC will be used in the method primarily for comparison between different periods’ average IOC with the English IOC. Another IOC comparison that is run is the comparison between the ciphertext IOC with the expected IOC of a period.

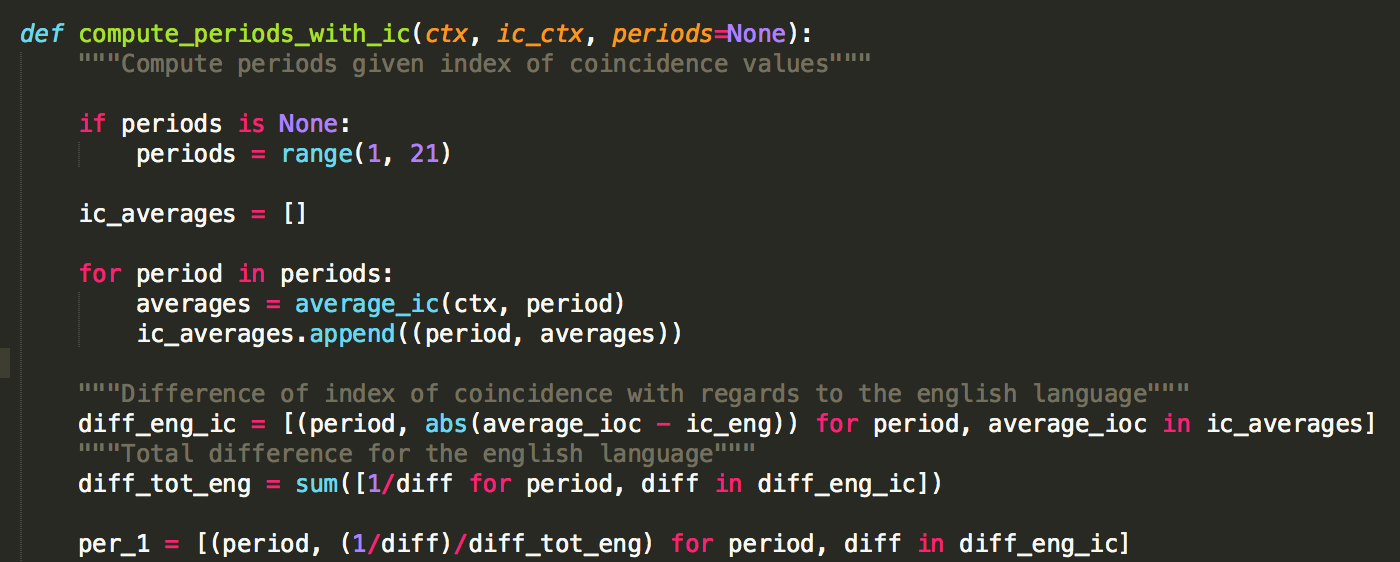


The function **ioc** is used to find the ioc of a ciphertext while **ioc\_subseq** is used exclusively in the Vigenere decryption since the input argument **subseq** is used exclusively in the Vigenere decryption process.

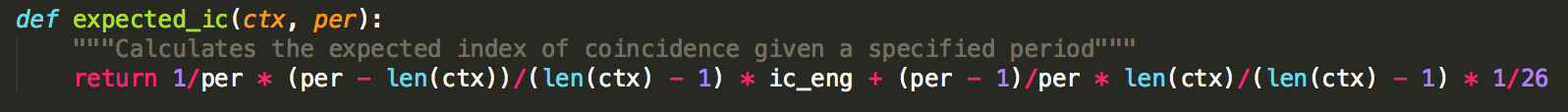
After obtaining the list of periods through the Kasiski method, the average IOC for each period is calculated. To find the average IOC, subsequences are extracted from the ciphertext in the following manner, with *p* representing a period value and *I* representing a subsequence:

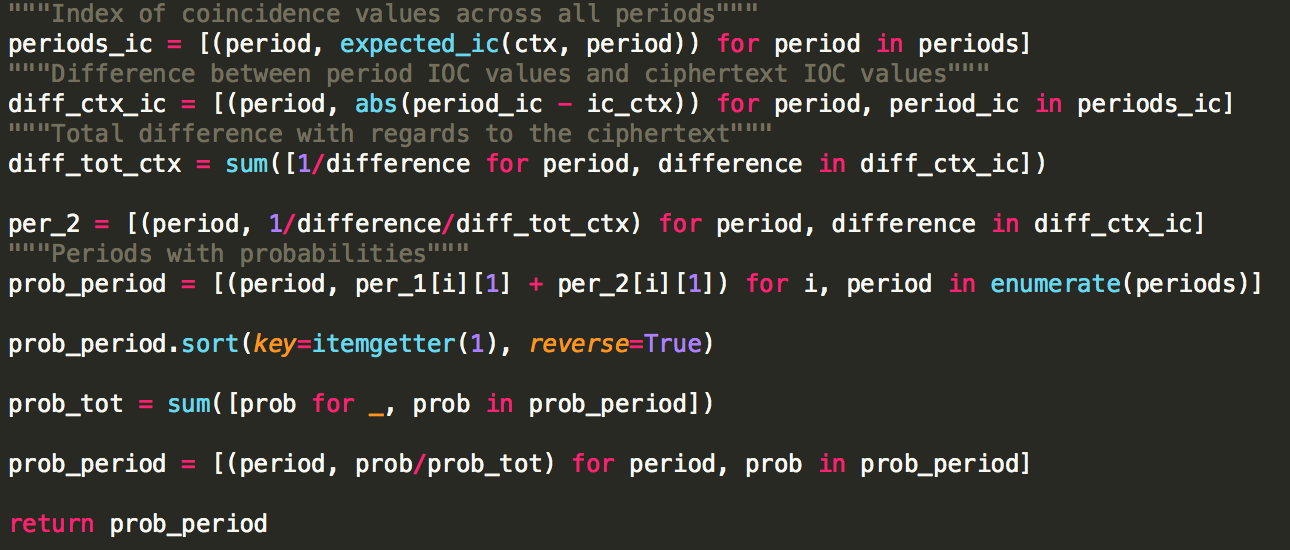




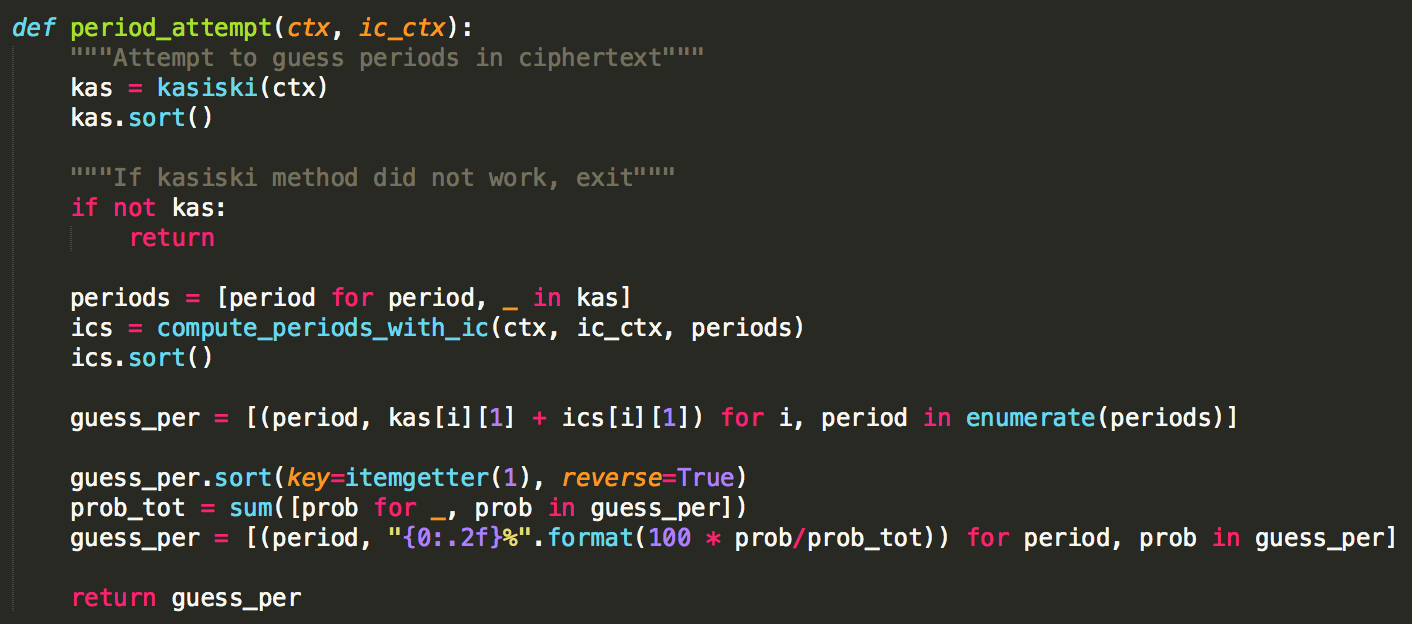
Once calculation of the average IOC is finished, it is then compared with the English IOC, since having an average IOC means that it has a higher likelihood of being an actual period for this cipher. Alongside the difference between each IOC, a probability of each period being real is associated with each one. To calculate the probability, it is done by dividing the inverse of the IOC differences and the inverse of the sum of the differences.****

The resulting variable from these calculations, **per\_1**, contains the corresponding probabilities for each period to make it easier to associate which probability is assigned to what period.

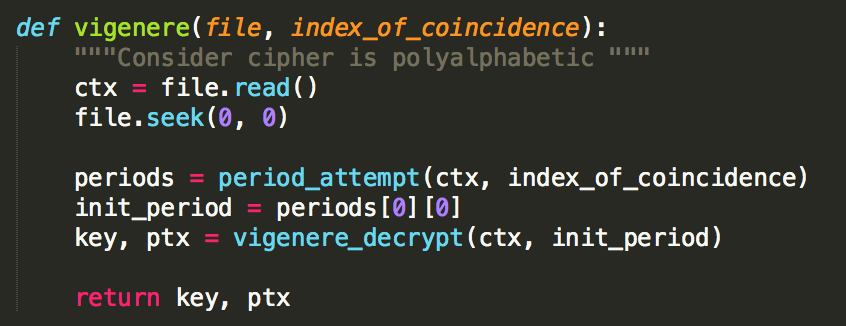
For the second part of the **compute\_periods\_with\_ic**, a couple more functions are utilized to compute the periods. First, **expected\_ic** is used to calculate the expected IOC value for a given period. The calculation process is illustrated in the return statement of the function.

Afterwards, the second part of the period computation is executed which involves doing similar probability calculations as the first part of the function, combining the probabilities from both parts to return a list of all of the periods alongside their corresponding probabilities.****

The function **period\_attempt** utilizes the computation of periods mentioned earlier and attempts to obtain the best estimate of a period. First, the function obtains the periods found via Kasiski’s method and the probabilities associated with those periods. After that, normalization of the probabilities occur alongside some modifications to make the probabilities to percentages to increase readability. Once that is finished, the function returns a list of up to 5 periods and its percentage of being a period of the Vigenere cipher.

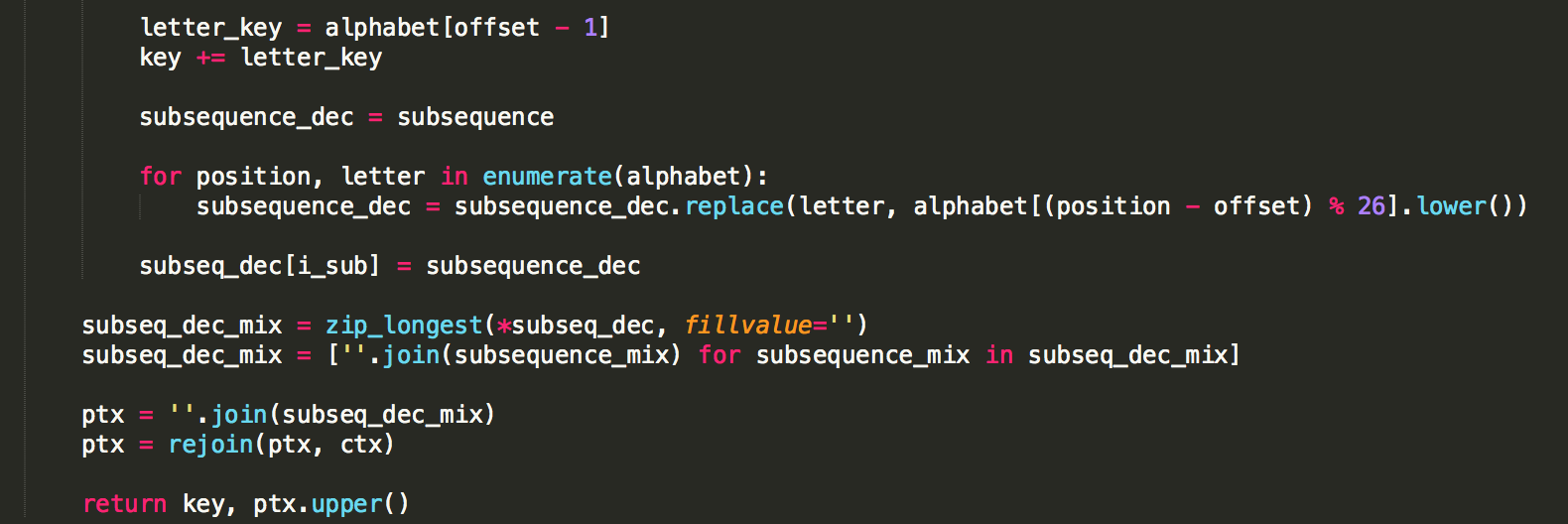
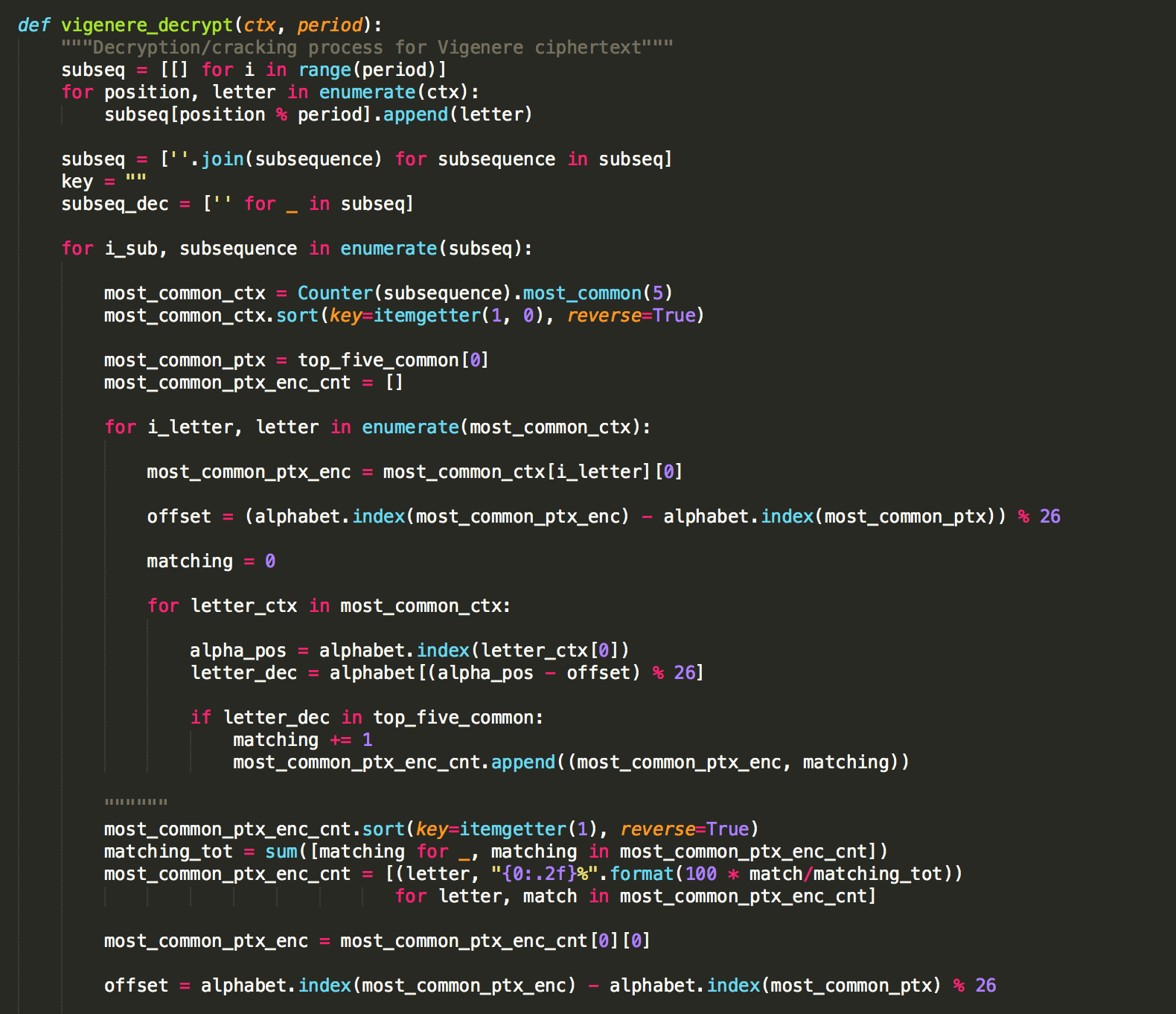
****

With this list, the first element is grabbed for decryption of the ciphertext in order to find the best estimate of the period.



Since the current use case of Vigenere is being only given the ciphertext, it is necessary to assume that the distribution of letters for each subsequence must be similar to the distribution of letters of the English language. This leads to the association of the most common letter of each subsequence to the most common letter of the English language.

The function that does all of the decryption is **vigenere\_decrypt**. First, this function assumes that the most frequent letter of the English language corresponds to one of the top 5 common letters of a subsequence. And the choice between the 5 letters of a subsequence is based off the number of iterations the letter has within the subsequence. If a letter produces x iterations based off the displacement obtained by that letter, then the subsequence can be deciphered to find its 5 most common letters, with which there are x letters that are in the top 5 letters of the English language. With this, the function will associate the letter with more successes as the letter that is most frequent in the English language, making it possible to obtain the displacement to decipher the subsequence.



**Shift cipher**