Although homophonic ciphers prevent the use of frequency analysis to decipher the encrypted message, we initally analyzed the frequencies of unigrams, bigrams, and trigrams from the encrypted text. Through the analysis of unigrams, we were able to make a reasonable association between the least common unigrams from the encrypted text and the 4 least common letters Q, X, J, and Z in the english language. Since these letters appear so infrequently, they appear just as infrequently in the encrypted text despite the frequency leveling caused by a homophonic cipher. In the encrypted text, we replaced the numbers with our corresponding guesses to make analysis of the encrypted text easier. After decoding Q in our encrypted text, we were able to make a reasonable guess as to what numbers were associated with the letter U since Q is most frequently followed by U.

After decoding five letters, we analyzed the encrypted text as a whole to find patterns that a bigram or trigram analysis may have missed. By analyzing the text, we noticed that the tirgram "4 # 52," started the text and occured frequently. We were able to make a reasonable guess that this trigram was associated with the word "the" since it is a common trigram in the english language and often begins sentences. After making these replacements in the encrypted text, we continued to make a few other random guesses and reasoned that the beginning of the text followed the same structure as other texts often provided in CS assignments. Under the assumption that the first few lines of encrypted text contained the title and copyright, we compared the encyrpted text to the shakespeare.txt file to decrypt the first section character by character. Since this was a large chunk of text, we had deciphered most of the encyrpted text and were able to isolate which numbers had not been deciphered. Since we were analyzing a text in which we replaced numbers with guesses, we were able to read through the encyrpted text and make reasonable guesses as to what the last few numbers decrypted into. We organized and consolidated our guesses and ran through the encrypted text to ensure we had not missed any numbers. When we confirmed that there were no longer any numbers in the encrypted text, we determined that we had decrypted the homphonic cipher.

The homophonic.py file includes methods which helped us get the frequency of certain n-grams of the either fully or partially unencrypted text file, as well as helping us to replace certain keys with certain values (found in key.txt)