The deciphered message is “FIRST I BELIEVE THAT THIS NATION SHOULD COMMIT ITSELF TO ACHIEVING THE GOAL BEFORE THIS DECADE IS OUT OF LANDING A MAN ON THE MOON AND RETURNING HIM SAFELY TO THE EARTH”

1. What difficulties did you run into writing the frequency analysis program? How did you resolve these problems?

I did not have too many issues with writing the frequency analysis program, and my initial plan worked. I did, however, have a few minor issues, like when I wrote “letterFrequency[i] = occurrence[i] / (double)text.length() \* 100” I initially did not put “(double)” before “text.length()” and for a while “return letterFrequency” returned an array with only “0.0”. I resolved this issue by printing out altered versions of the code, such as “letterFrequency[i] = occurrence[i] + text.length() \* 100” and noticed that certain altered versions did not have the issue of returning “0.0”. After a while, I finally realized my problem and added “(double)”. Another minor issue that I had was that I wrote “%3s %17f %24.2d\n” instead of “%3s %17d %24.2f\n”. It did not take very long for me to remember that d was used for integer values and f was used for double values.

2. How closely did the frequencies of letters in the plaintext and the ciphertext correspond? What could be done to improve the accuracy of the correspondence?

The frequencies of the the letters in the plaintext and the ciphertext corresponded quite closely usually (I would estimate below a 0.5% difference in most cases). The accuracy of the correspondence can be improved by recording the frequencies of larger text samples in both English and the ciphered language.

3. This assignment involved writing a program and then interpreting the results the old fashioned way. Propose an extension to this project that would allow the computer to do all the work so that you only have to evaluate whether the decoded message makes senses. You don’t need to actually write another program to do this, just consider how it could be done and clearly describe your solution in a well written paragraph.

There are a few helpful extensions that could be added that would decode messages rather accurately. One helpful extension would use statistics, perhaps standard deviation, to determine what percent differences between the frequencies of corresponding letters would seem too large to match the letters together. This extension would eliminate pairing letters with large percent differences and would tend to match letters with lower percent differences. An additional extension would record entire words with large frequencies and try to match them with common English words with the exact same number of letters. Some examples of common English words would be “the,” “this,” “that,” “a,” and “I.” These words would be compared to ciphered words, and patterns would be analyzed. For example, when patterns are analyzed, “that” and “this,” common four-letter words, would be clearly distinguished from each other since “that,” unlike “this,” starts with the same letter and ends with the same letter. The extension would make sure that a four-letter cipher text that starts and ends with the same letter does not correspond to “this,” and it would increase the likelihood of matching it up with “that.” Through this process, the extension may narrow down the cipher letters that correspond to the English letters of “this” and “that” with high accuracy. Another extension would be one that would tend to eliminate extremely unlikely letter combinations. For example, in English, no real word juxtaposes “q” with “z.” This juxtaposition would only tend to occur in rare usernames, texts where the users intentionally banged their keyboards, possibly other foreign languages, etc. If a program found an unlikely combination of letters, such as “qz,” it would try to either change the letter q, z, or both to different letters with similar frequencies. A final helpful extension would be one that would compare deciphered words to words in an English dictionary and a “name” dictionary. If most or all of the words match up with real dictionary entries, the program can be confident that it deciphered the text accurately and would not have to match ciphered letters with other English letters all over again.