**RESULTS**

Original Alphabet: abcdefghijklmnopqrstuvwxyz

Cipher Alphabet: KMGVZOTQACESNWYBDXFIHPJLRU

Ciphertext Message: OAXFI A MZSAZPZ IQKI IQAF WKIAYW FQYHSV GYNNAI AIFZSO IY KGQAZPAWT

Plaintext Message: first i believe that this nation should commit itself to achieving

**REFLECTION**

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

At first, the main problem that I faced was decrypting the message itself. The frequencies didn’t help me much, as I believe my writing style was too different to that of the target ciphertext. To resolve this issue, I manually tested out the most obvious matches from a superficial observation (like z -> e from cipher -> plaintext) and then used the frequencies of the remaining 5 or so to deduce the rest.

1. 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 correspondence between my set and the cyphertext was high, but not enough so to be good enough to make immediate correlations between the ciphertext letters and their plaintext counterparts. To improve the accuracy, a higher sample of ciphertext and plaintext would probably help in showing the ratios of each letter to the total number of letters in their set.

1. 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 must evaluate whether the decoded message makes senses. You don’t need to write another program to do this, just consider how it could be done and clearly describe your solution in a well-written paragraph.

While working on this project, I determined that it would be more time-effective to create a program that would automatically decrypt an encrypted message given that the user had the decryption alphabet already (cipher-letters -> plain-letters). Additionally, I thought of creating a separate class/method that would go through all possible permutations of the alphabet to brute-force the key rather than heuristically analyzing the message given letter frequencies and manual checking. However, I found that this would be too time-consuming for the scope of the project and how far I wanted to take it. That being said, I believe that it is more than possible for a program to be created in Java that will brute-force the key that can decrypt the ciphertext to plaintext given enough time and computing resources.