Project 1

Procedure:

1. Use the SplitString.cpp I’ve attached to split the original cipherText.txt
2. With each split use the provided FrequencyAnalyzer.cpp to check one of the split files and compare it with the letter frequency of the English language and look for similarities, if it doesn’t match give or take a percentage then split the original file into more pieces than the last time.
3. Once the frequency matches check all the other split files frequencies to confirm and if they all check out then congrats you’ve found the key length. In this case it is 5.
4. Once you know this its time to decrypt, so analyze the frequency of each split file and however far the main letter (in this case E) is from the letter that currently has its normal frequency according to the English language frequency chart, that number represents the index of the key for this file.
5. Decrypt that file with the given key check the decrypted file with the frequency analyzer and check the frequencies line up accurately to the English frequencies and repeat for the other split encrypted files.
6. Once this is all done you should find that they key is ironically SONIC, must be because when it came to decrypting this file it can be done fast, like sonic.
7. Finally use that key to decrypt the original non-split cipherText.txt and boom you’re done.

(decryptedText.txt also attached.)