Ben Silbaugh

Network Security

Mono-Alphabetic Analysis

I first started to decrypt the text file by setting my frequency directly mapping to the frequency of letters in the English language that was provided in the slides.  I.e. most frequent was set to ‘E’ and so on and so forth. I did this by pushing each character of each line through a switch case and printed the key instead of the char itself.  Once I printed the entire document I could find almost complete words, and switch letters around, and rinse and repeat this process until the entire document was decrypted.

I have a lot of methods that go unused, as I was trying to automate the process, but then I gave up. I planned to do this by assigning each character a Boolean value, and when a jumble process created a full word (as according to a dictionary) I would set all those characters to true and stop jumbling them. But I ran out of time and energy to finish it.

The plaintext is in the output, runs in NetBeans. Unsure if it will work in Eclipse, as Eclipse gets mad when you do multiple things in one line.

BONUS:

For the bonus question I had a much harder time decrypting it. This was because normally you look for almost completed words and figure out what to change, but the number of words here was so small many were very hard to figure out what word they were supposed to be. Also the letters were in much closer frequencies so that made it harder as well. The only way to solve this was to guess and check many times.