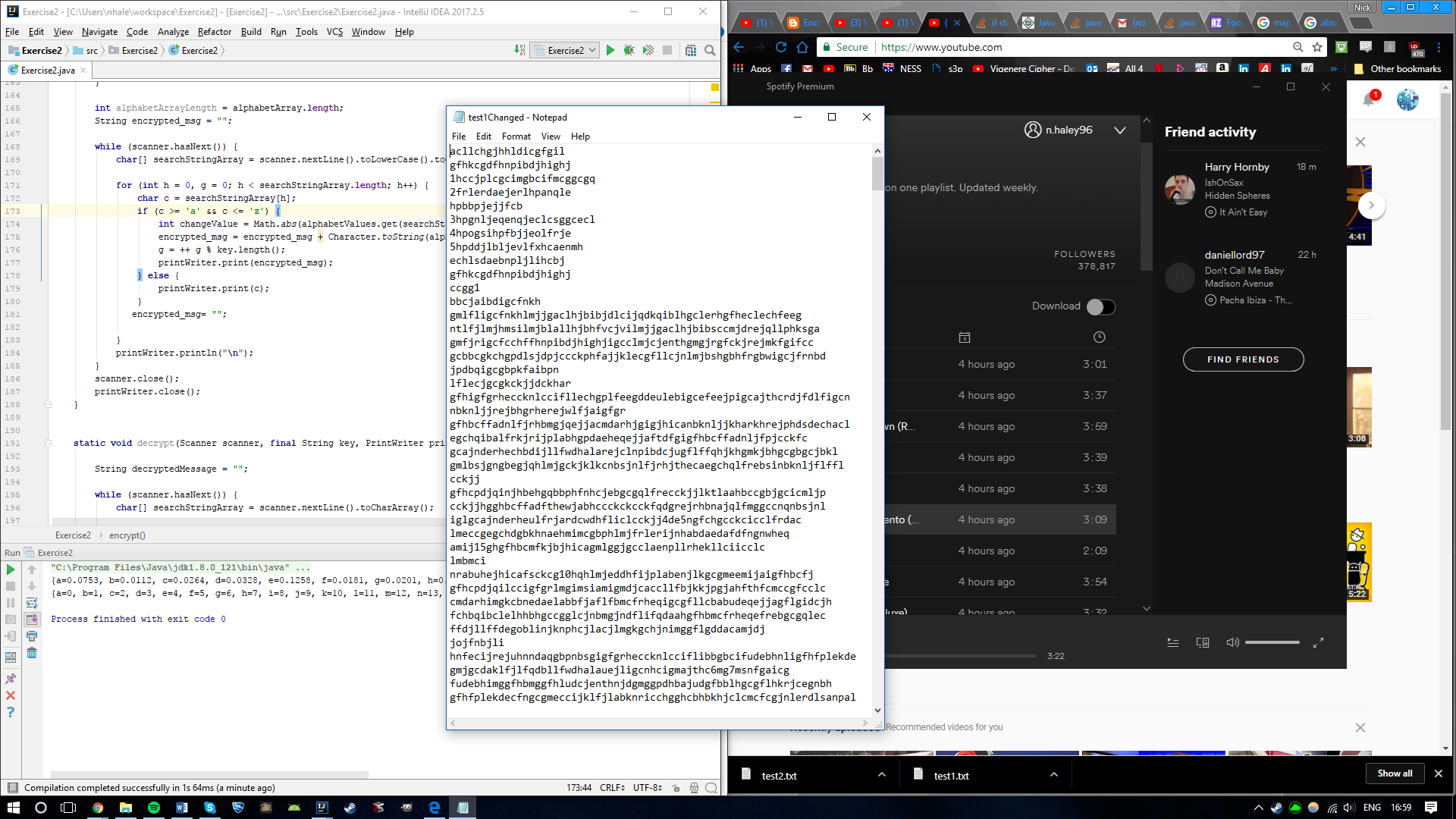
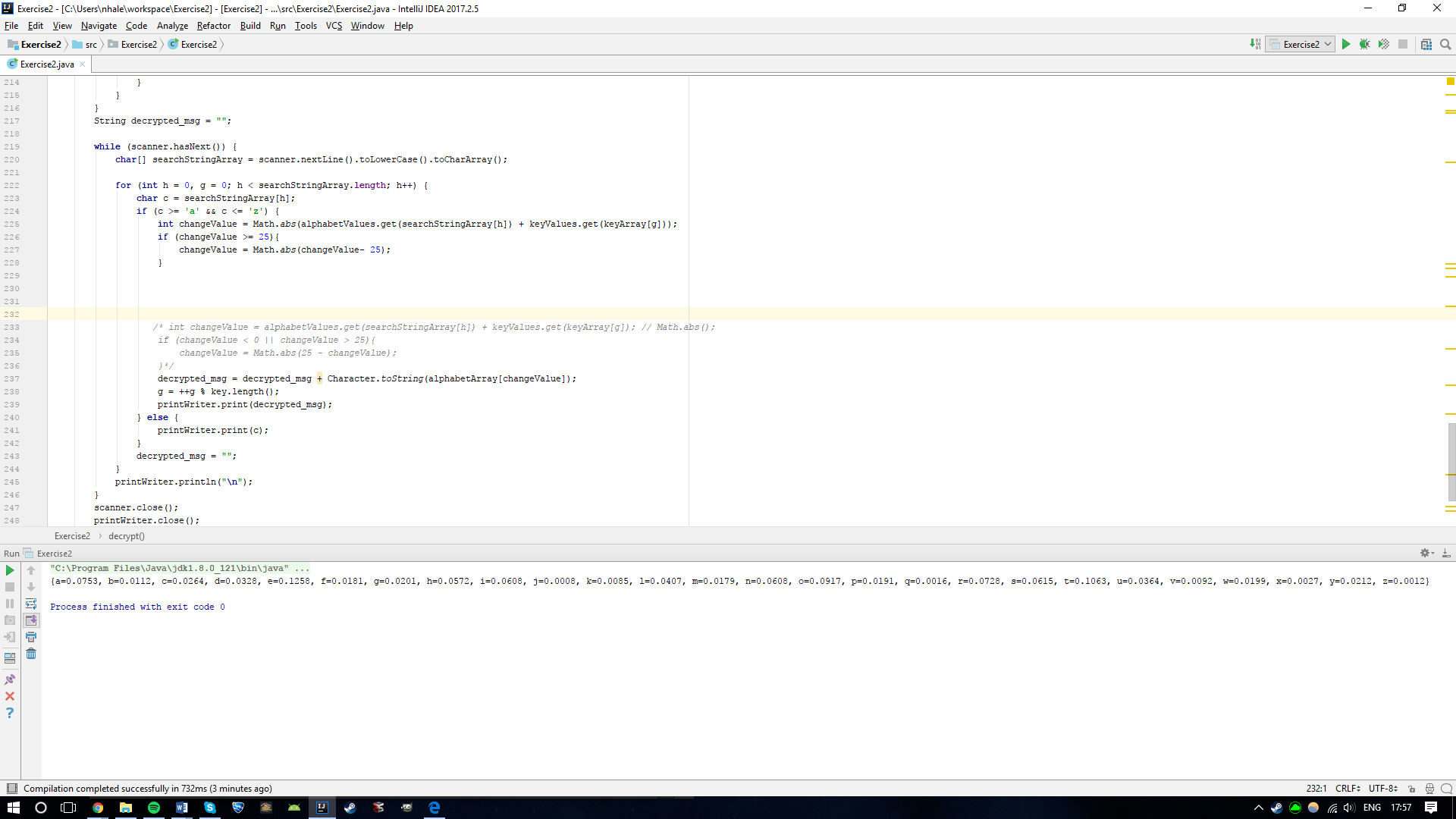
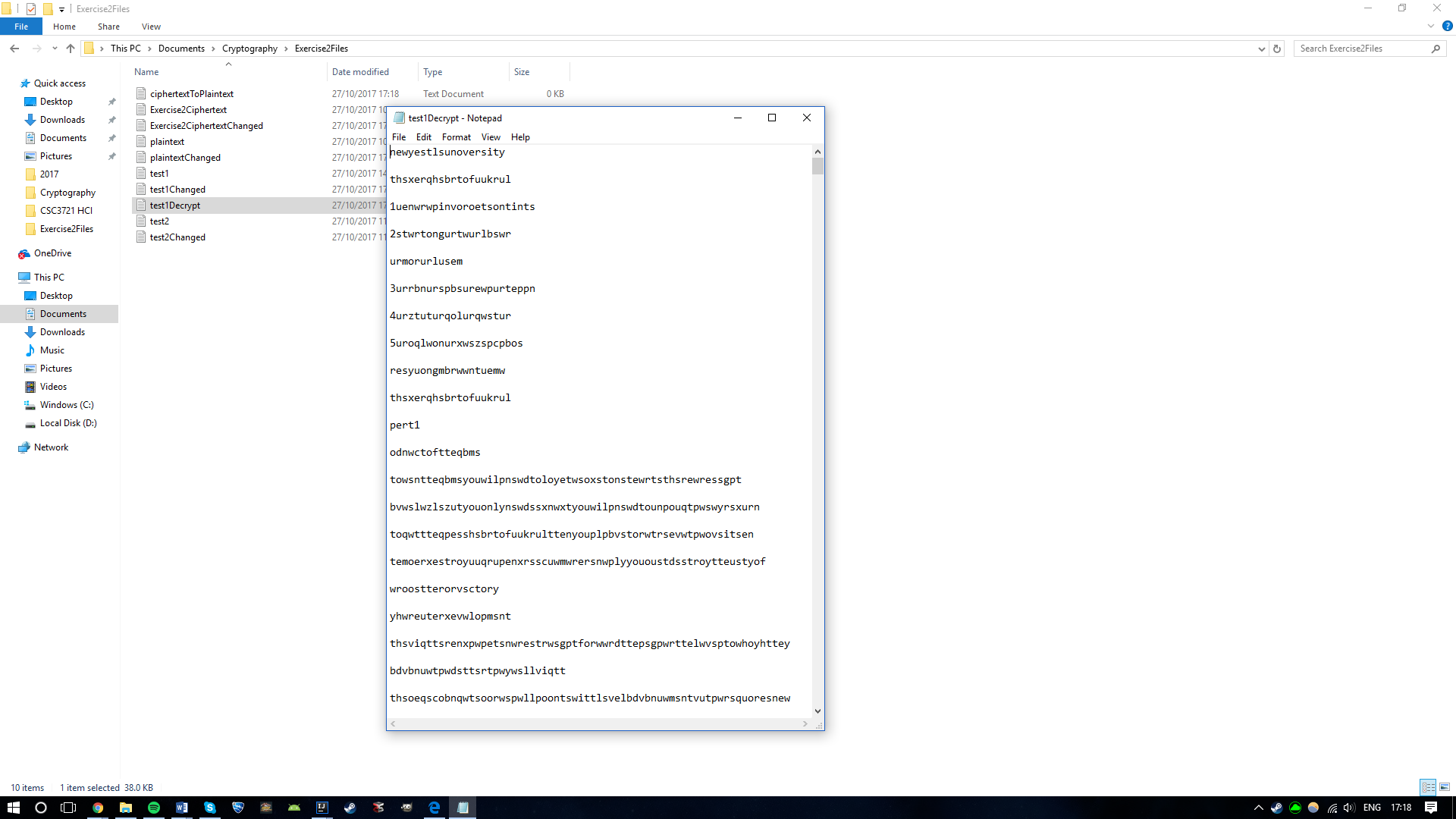
First line in this file is “newcastleuniversity” encrypted using the key “ncl”.

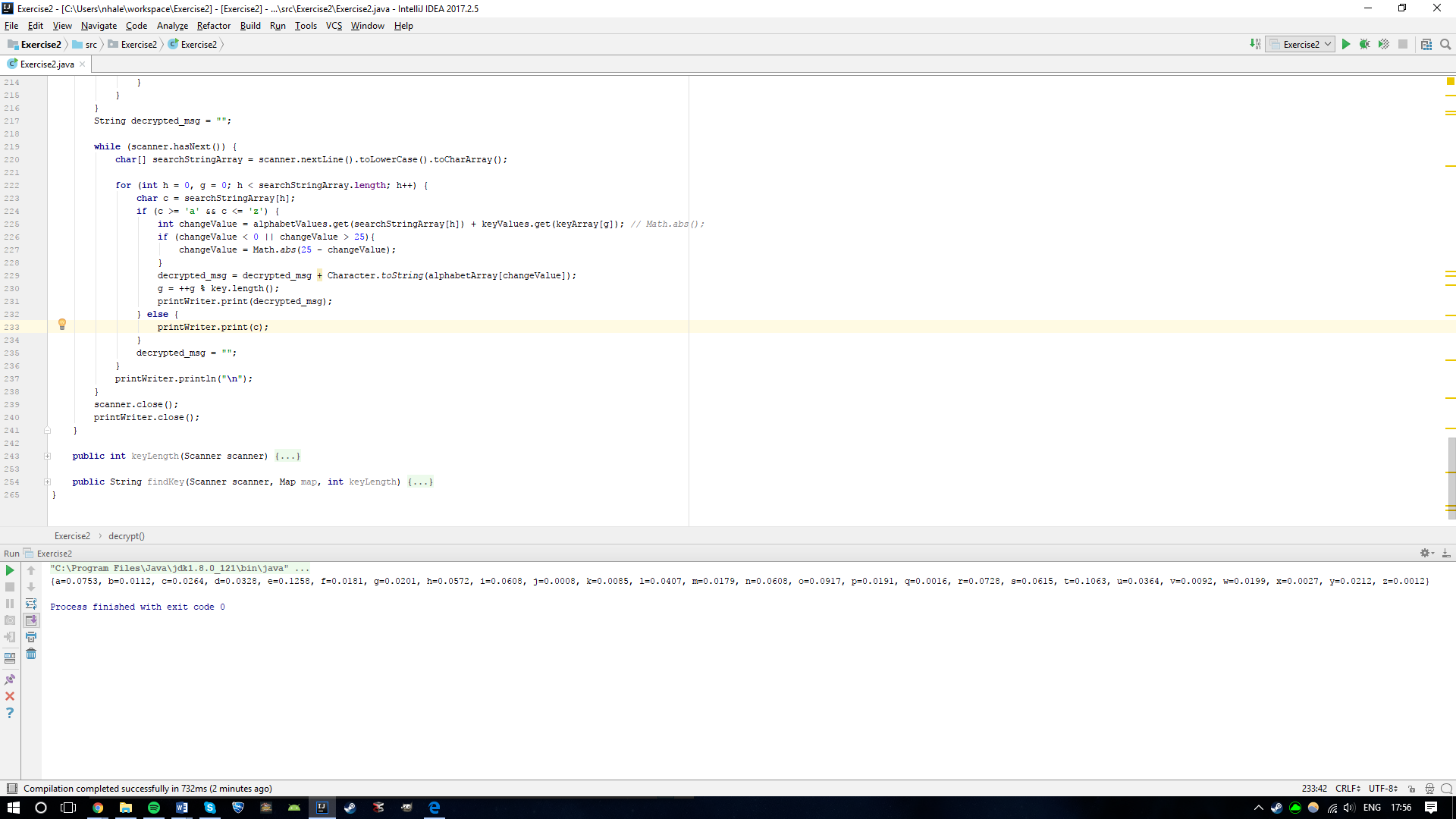


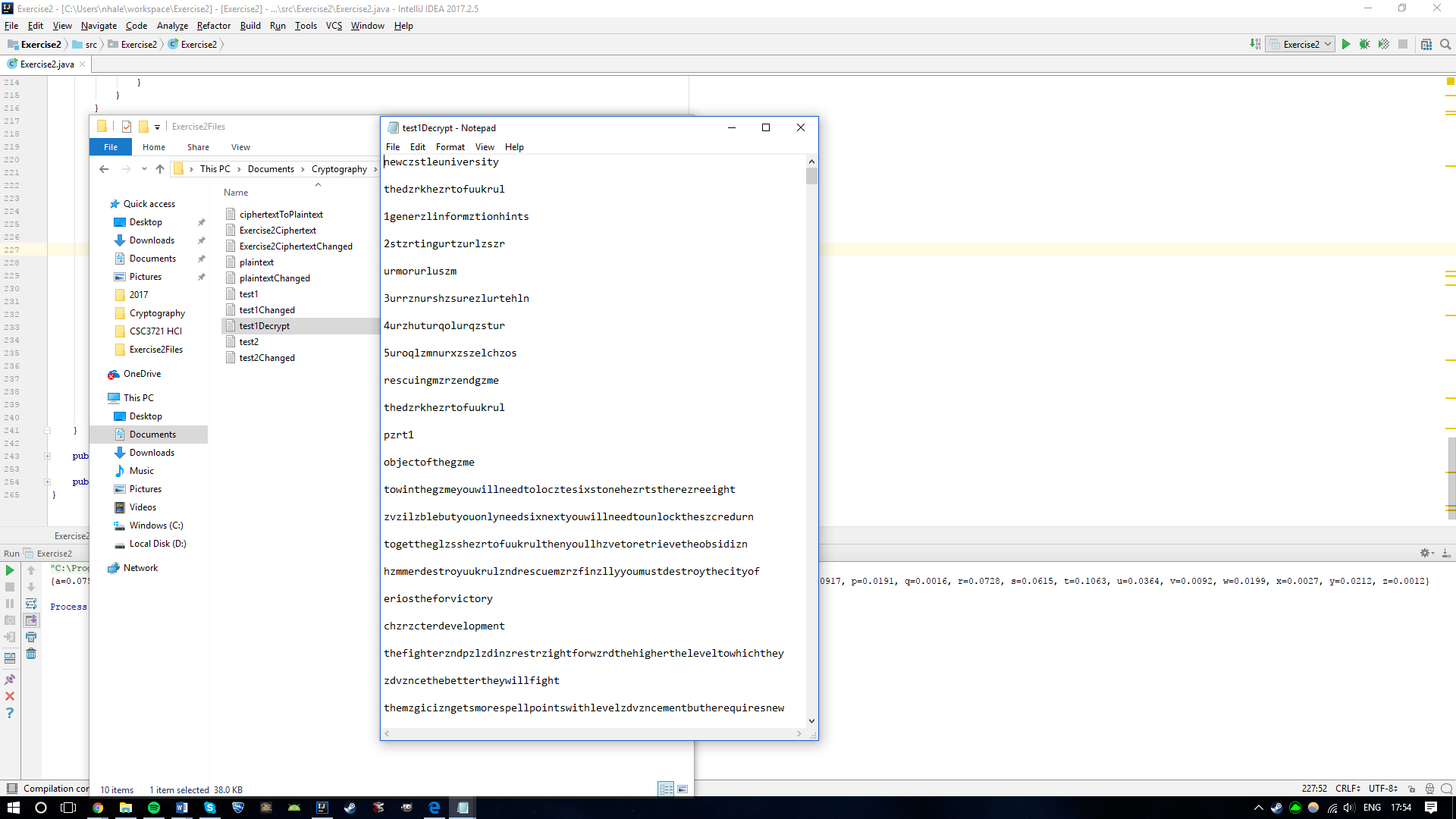
First attempt to decrypt: It is close to “newcastleuniversity” but some characters are wrong.

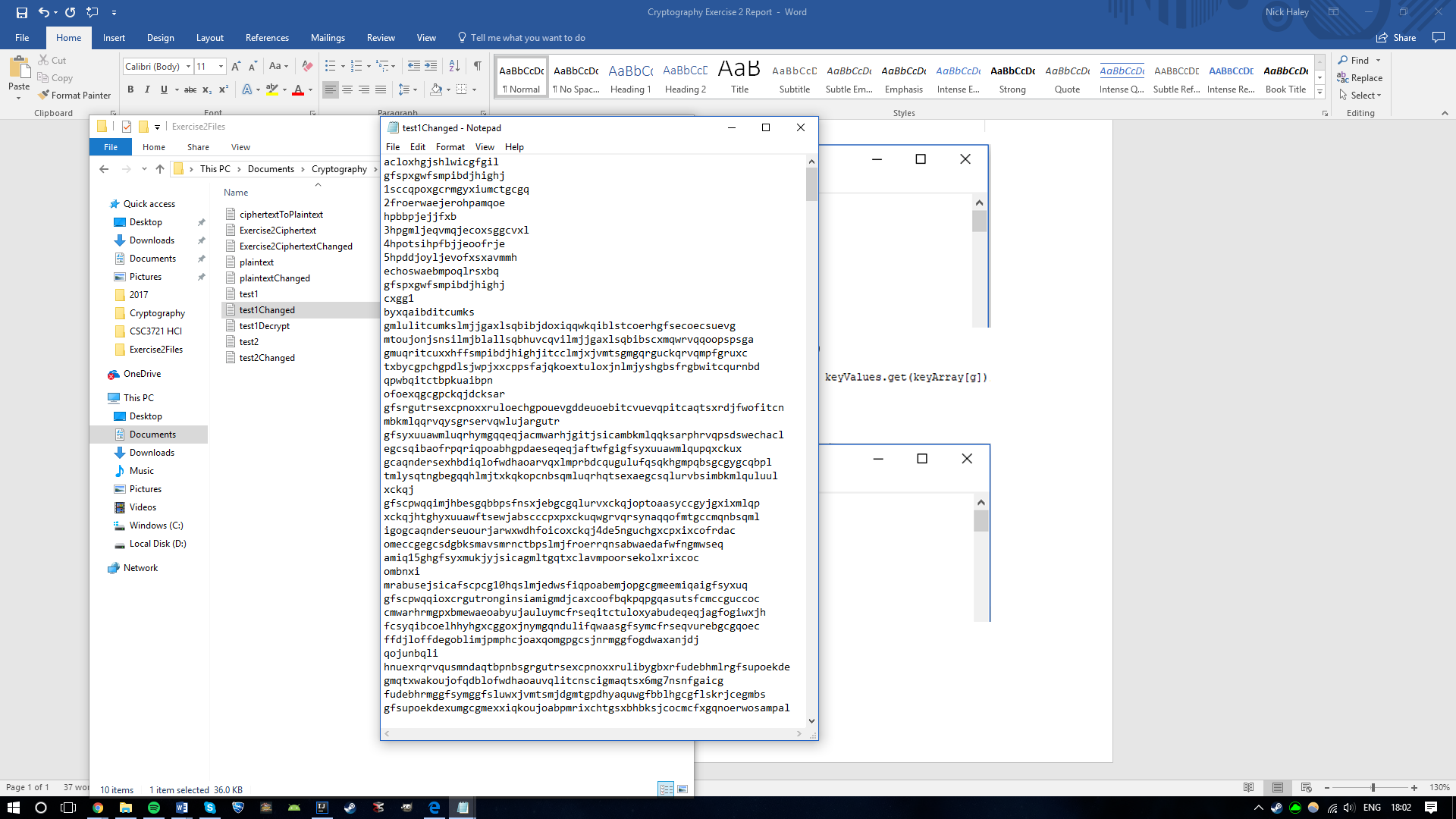




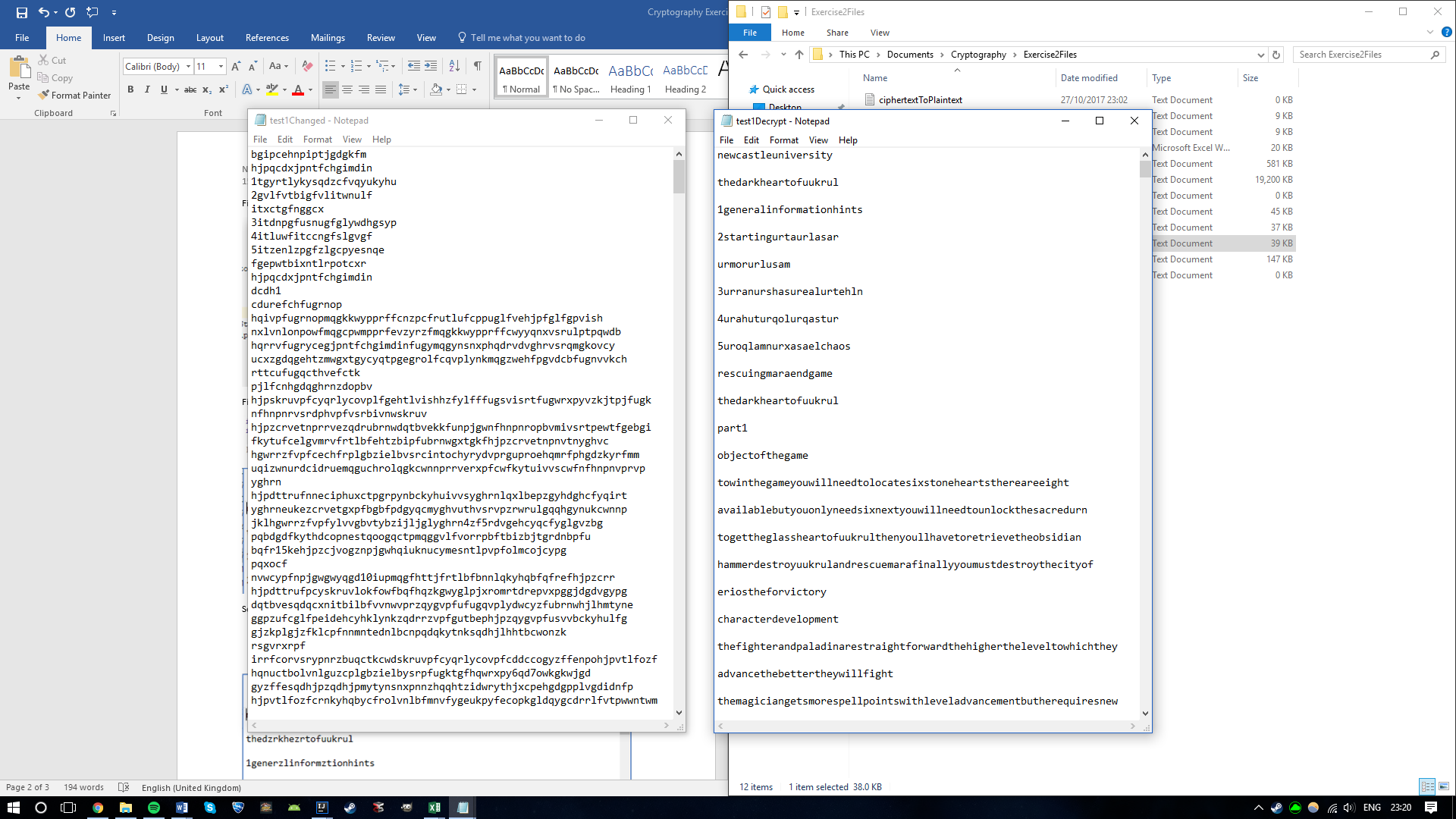
Second Attempt: Ambiguous when decrypting back to an “a”.

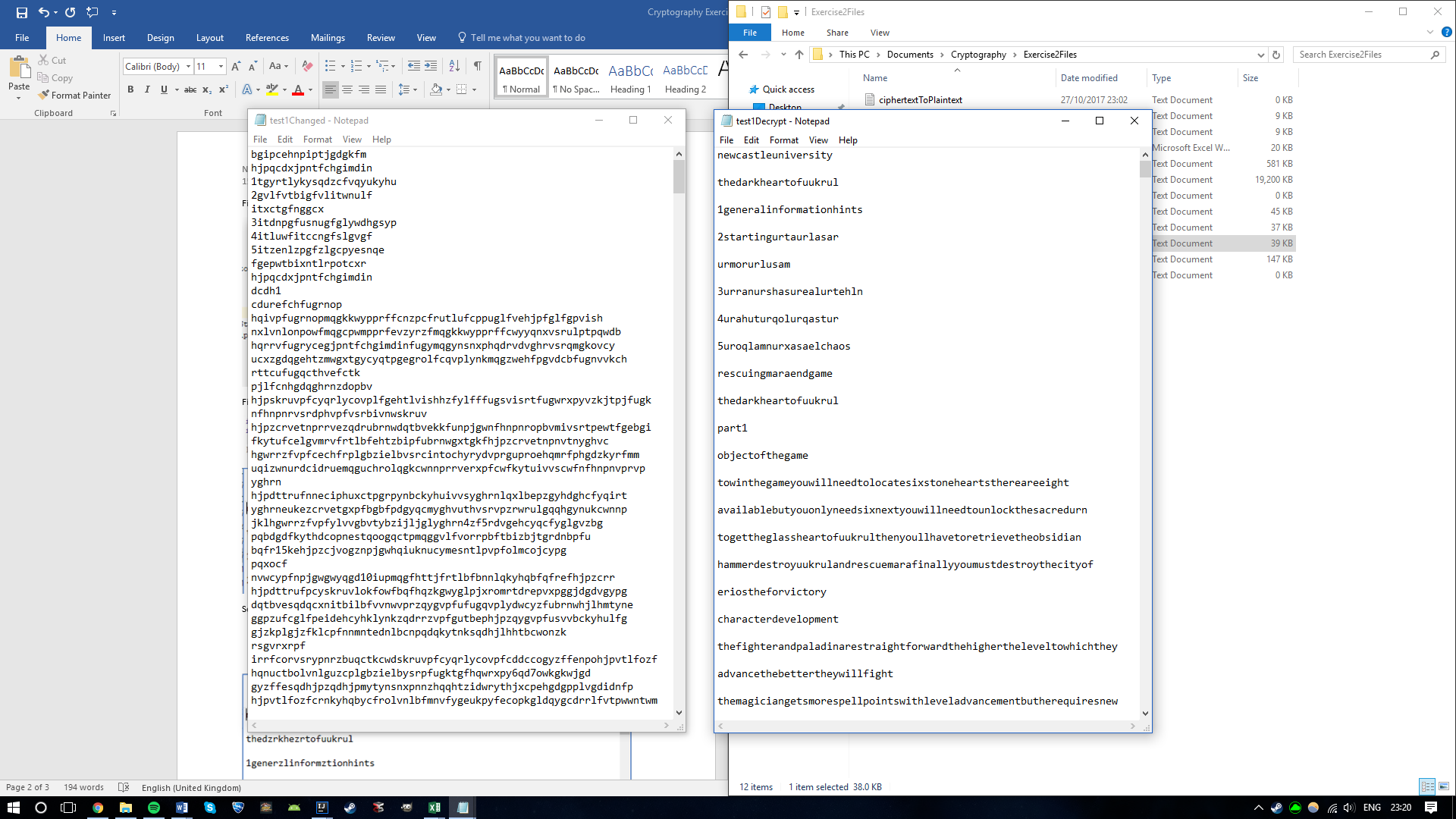


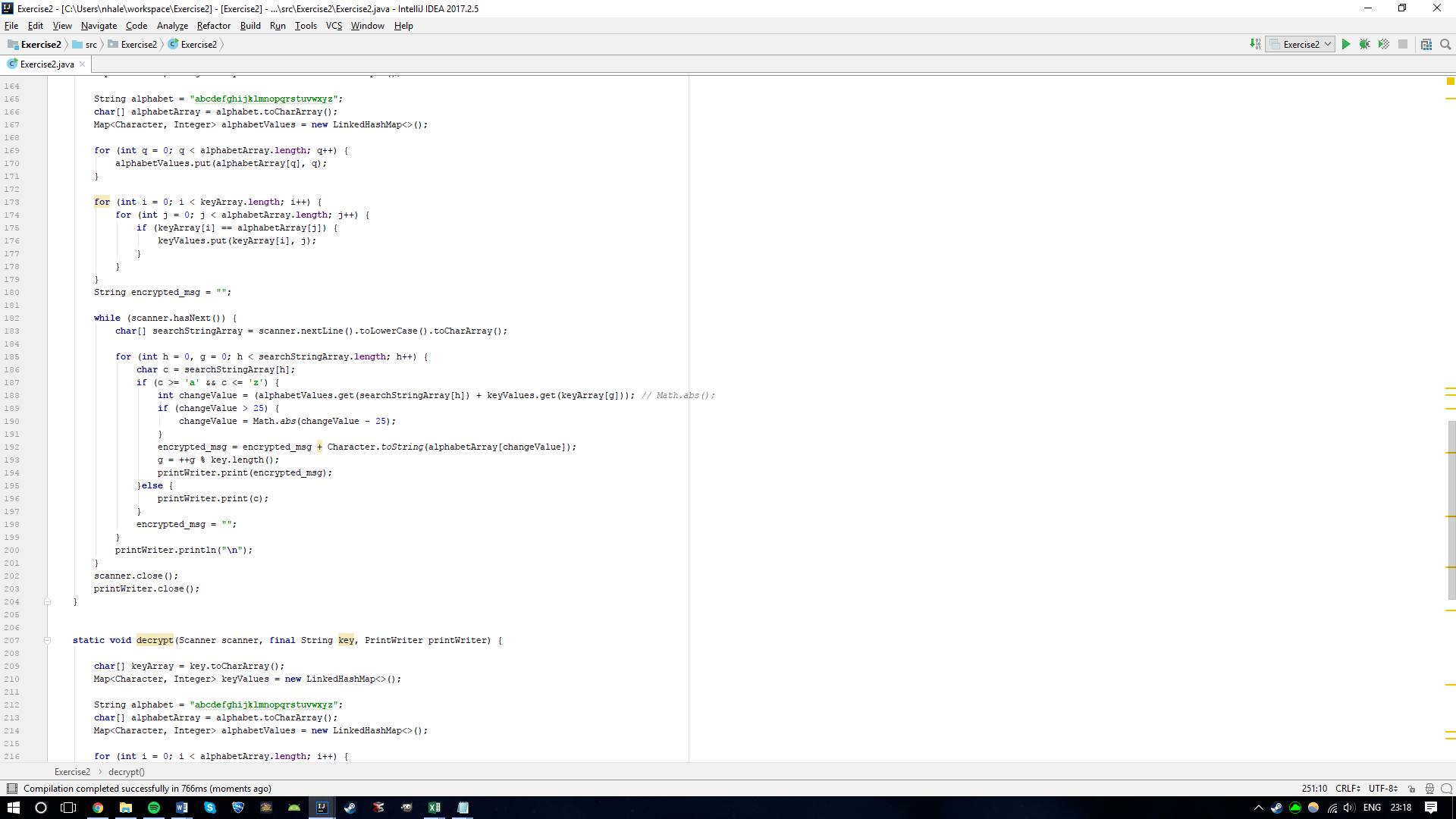




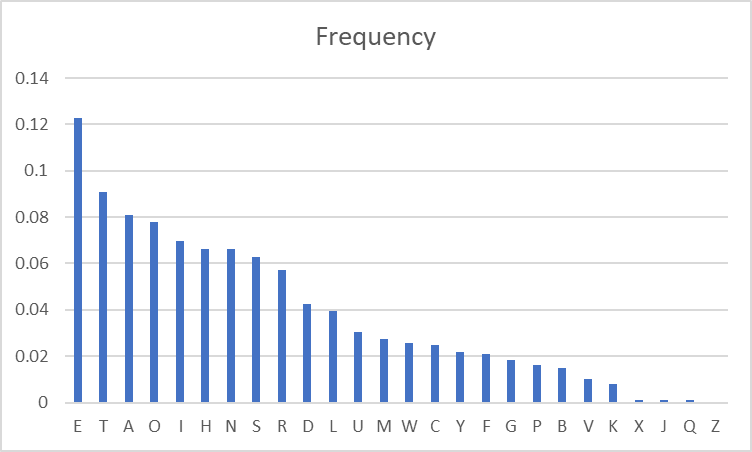
The final problem I faced was dealing with the letter ‘a’ in the plaintext. In the encryption step when you subtract the value of the key letter from an ‘a’ character the result will always be negative. Therefore I used the Math.ads() operation to find the absolute value. However I realised that my encryption and decryption calculations were the wrong way round. So I swaped them and the encryption worked immediately



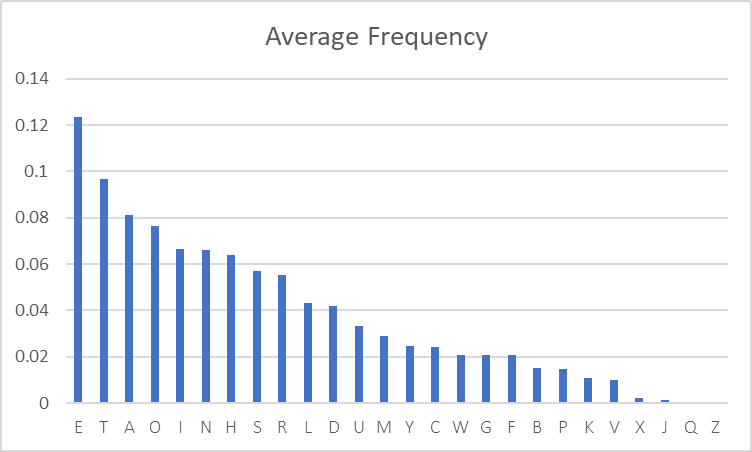




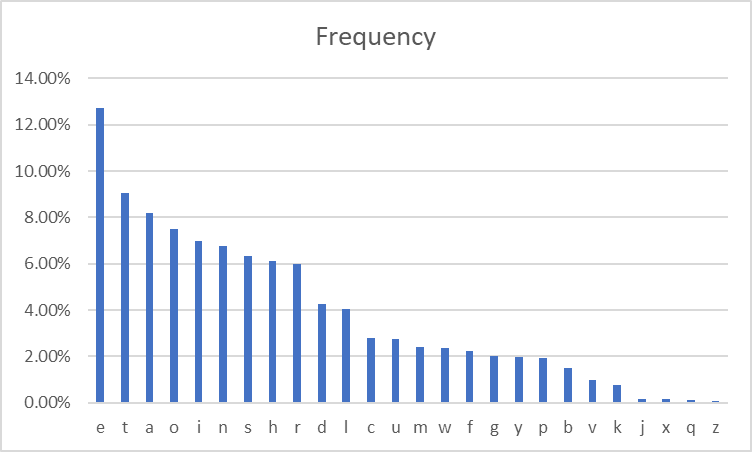
Plaintext Frequencies:



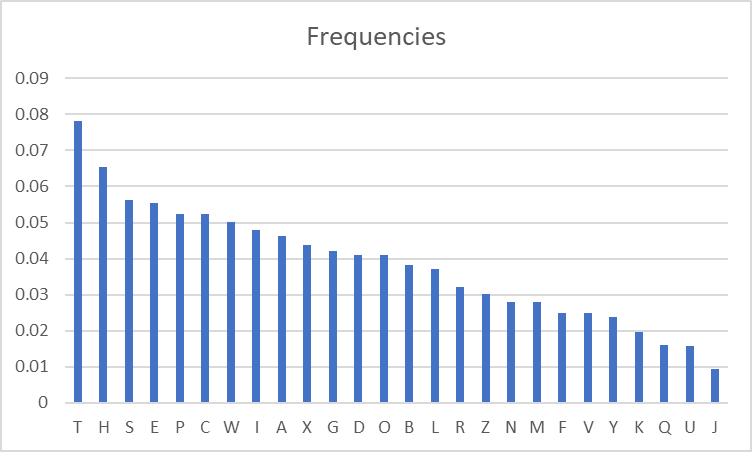
My calculated average Frequencies:



English Alphabet Frequencies from Wiki:



Cyphertext Frequencies:



The frequencies after encryption via the Vigenère cipher are far more even and less extreme. This is because there is each letter is not simply swapped for another via a shift of the sequence. The letter choice is based on a key and each part of the key could be any letter in the alphabet. This means that the encryption of a letter is far less predictable which will make the cipher harder to break.