**Polyalphabetic cipher**

The polyaplhabetic cipher program I have written is based on the Vigenere cipher. I take a text file and a string of letters, or password, and shift each letter of the text according to the letter of the password it is aligned with it is aligned taking into account any spaces or punctuation.

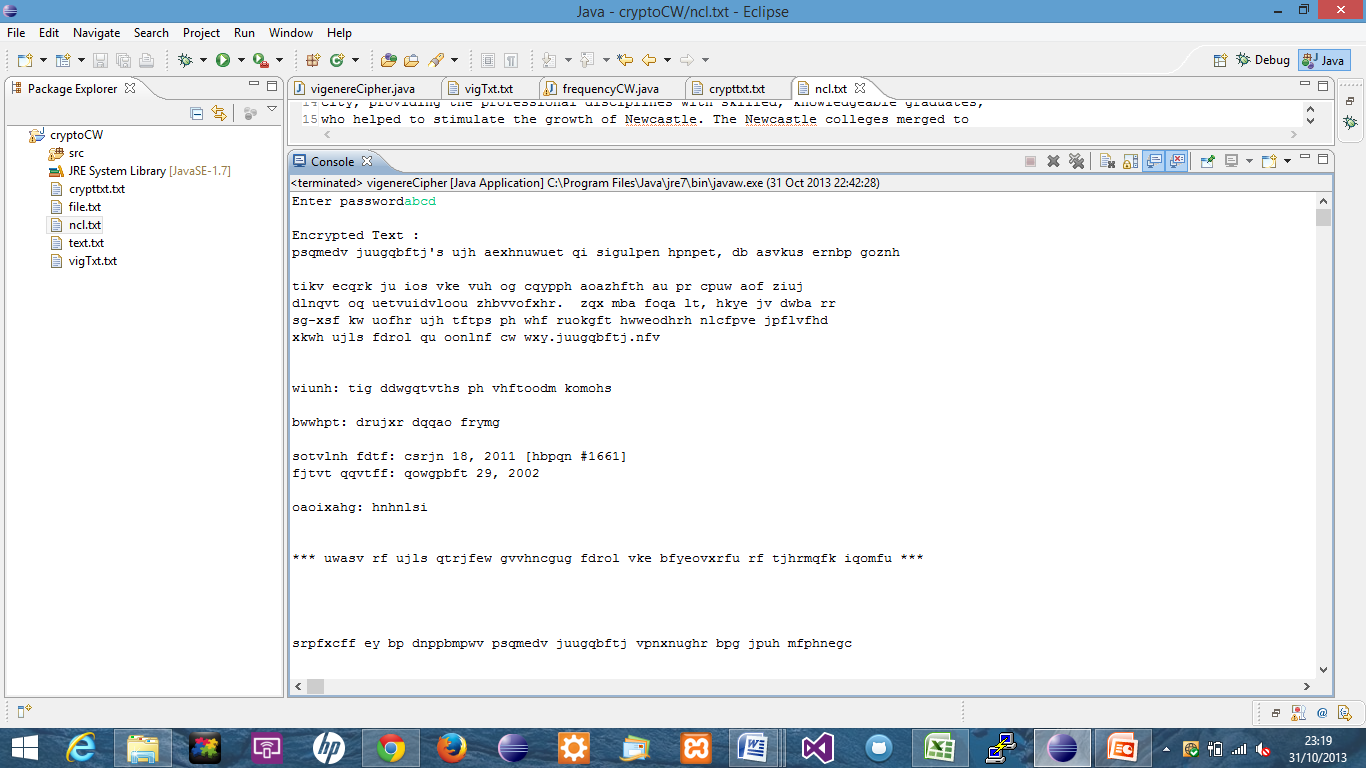
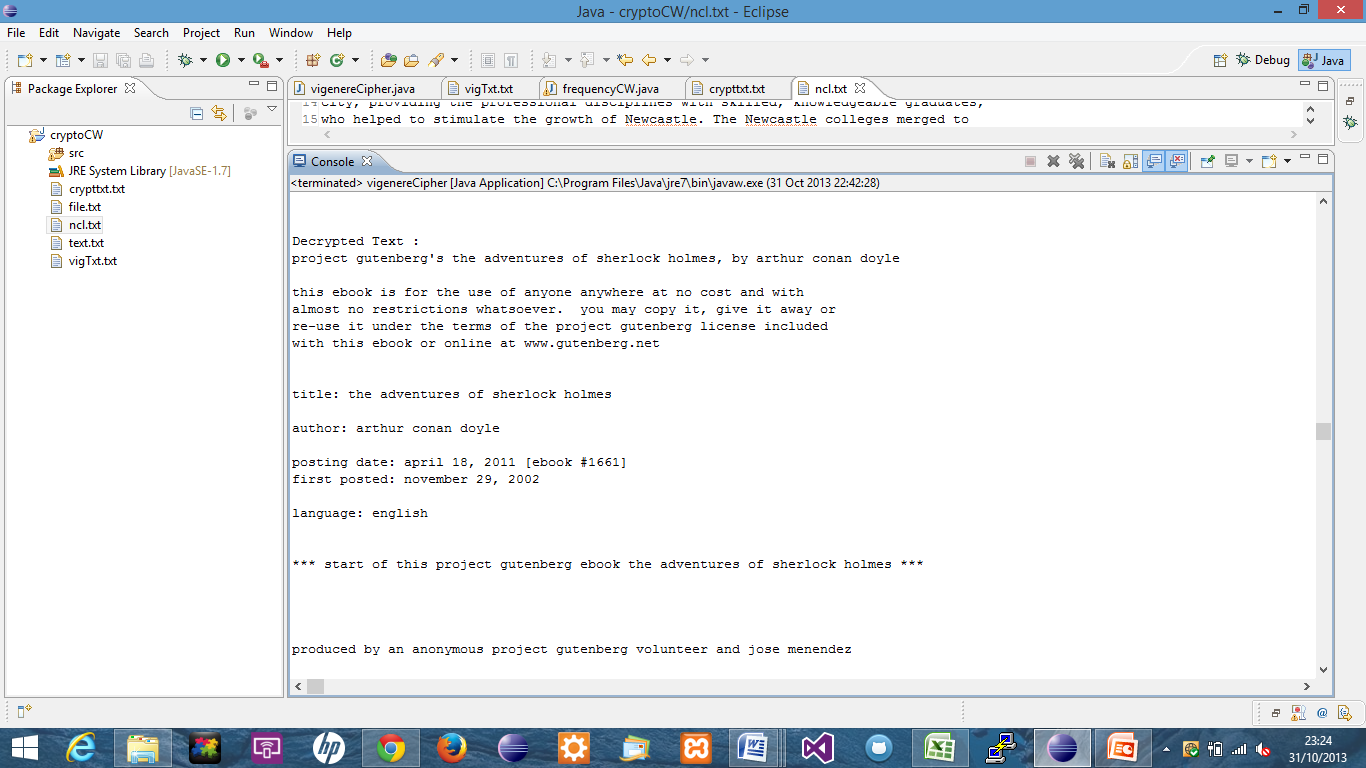
For example

PW abcda b cd abcd abcd

Text hello, I am some text

Cipher hfnoo, j cq spoi tfzw

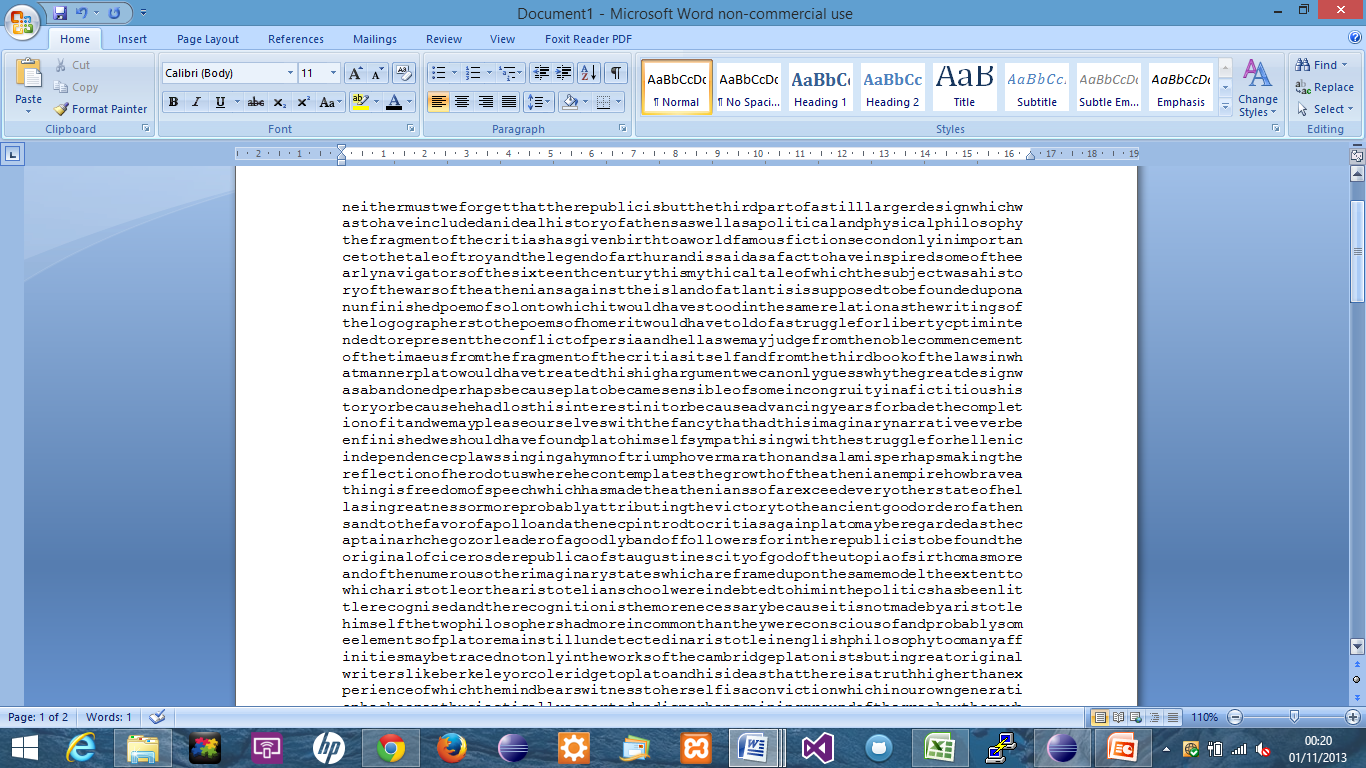
The same password was used to encrypt and decrypt a large text file, the results are shown below:



When analysing the letter frequency of the cipher text, it becomes clear that there is not a dominant frequently used letter. There is less use of some letters than others, but not as much as the most commonly know letter frequencies.

To break the cipher text, I took the file and analysed the frequency of trigrams. I did this as I felt that I would more than likely find common three letter word repeated like ‘the’ or ‘and’. I found the most frequent occurring trigrams in the cipher text are ‘hwp’, ’mvt’, ’tas’ and ‘ehx’. I took the most frequent ‘hwp’ and fed that back into my program to find the key length.

To find the key length I used the Kasiski test. I took the first three occurrences of the most found trigram and calculated the distance between the occurrences. I then used these distances to calculate the greatest common denominator. The GCD will be used to determine how many columns I would split the cipher text into. Because I wasn’t sure how to automatically create the desired number of arrays to split the cipher text into, I did this manually. Once I had created five char arrays, I sorted the cipher text in to them. I used a counter and if statement, as I was having problems using multiple for loops, I used the counter to determine which array each letter was sorted into. I then took the first array and found the most frequent letter occurrence, I made an assumption that the most frequent letter is ‘e’ and using the ASCII value (e =101) I subtracted e from the maximum letter found and added the ASCII value of ‘a’ to find the first letter of the key. I continued this until all letters of the key had been found, and returned the key (plato) to the decrypt method.



(please note, my console would not show the text for some reason, though I could copy and paste it into word.)

Times higher education. (2013). *World university rankings.* Available: <http://www.timeshighereducation.co.uk/world-university-rankings/2012-13/world-ranking/institution/newcastle-university> Last accessed 31/10/2013.

*Kasiski Test.* Available: <http://www.personal.psu.edu/users/m/r/mrk5094/Kasiski.html> Last accessed 25/10/13.