**Mono-alphabetic Substitution Cipher (Simple Substitution Cipher) Report**

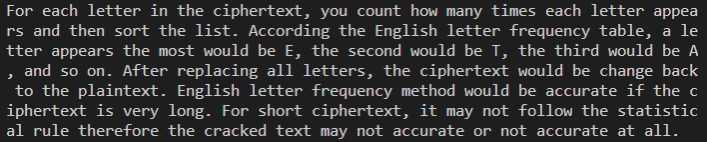
*Student: Lu Lin, Feb 10, 2019*

*Computer Science, Columbus State University, Spring 2019*

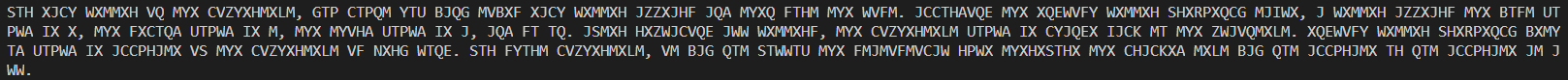
The mono-alphabetic substitution (also called simple substitution) cipher is basically a permutation of the alphabet, the permutations are one-to-one, and the substitution is invertible. In my assignment I substitute plain alphabet as the following table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| plain | A | B | C | D | E | F | G | H | I | J | K | L | M |
| cipher | J | I | C | A | X | S | E | Y | V | D | K | W | B |
| plain | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| cipher | Q | T | Z | R | H | F | M | P | N | U | L | G | O |

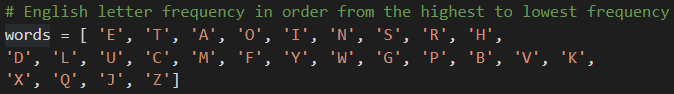
Using this table, I could encipher any plain text (after removing space) to cipher text. For example, I now just write the plain text as following:

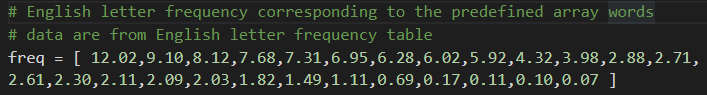


And then get it enciphered using my program, here I got cipher text:

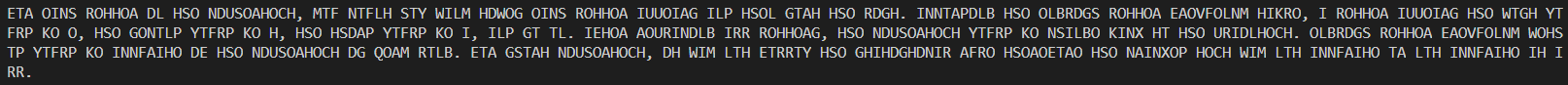


The second part of this assignment is the Cryptanalysis. To implement it I use two methods: English letter frequency method and hill-climbing method. For the first method, I use the English letter frequency table as given in the assignment instructions, in my python program I instantiate a word array and a frequency array as following:

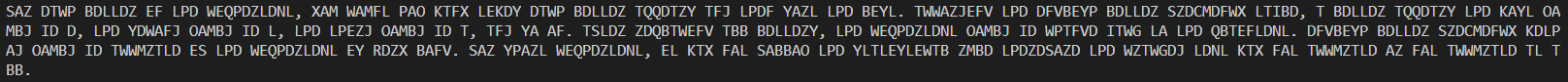




For decryption, I use the previous ciphertext as an example. In my program I calculate the frequency of English letters in ciphertext first, and sort and store the frequency in order of the most frequent to the least frequent. Then I match letters to existing English letter frequency array. For this step, I match words with the closest frequencies. Since this is a one to one relationship, if one letter in English letter frequency array was matched, the second closer letter would be matched; if the second closer letter was matched, the third closer letter would be chosen…In this way my program got the decrypted text as following:



As described in assignment instruction, the longer ciphertext would produce more accurate result because the words frequency would be closer to English letter frequency. Since the input test text is very short, the descripted result won’t be as accurate as expected. Hence, I use the second method for decryption: hill-climbing algorithm. Using the fitness calculation formula as in the assignment instructions my program calculates the fitness of various keys repeatedly to find a better key, as suggested in assignment instructions, the program runs 1,000 iterations to give results. Using the same cipher text as in the first method, hill-climbing method get the deciphered text as following:



Both methods have the same issue: the short cipher text might not get accurate result. To improve the decryption result, I would do two more things to make it closer to original plain text: find a much longer plain text for cryptanalysis and run the hill-climbing program 10,000 iterations or more (because of time constraint I will do it later).