Lab02 - Classic Cryptography - Monoalphabetic Substitution

## Part 01)

1. **Submit your commented code from Part one as “Lab02\_part1.py” to canvas. It needs to be documented and it needs to run correctly.**

(30 points)

1. **What is the plaintext message?**

(10 points)

ONCEUPONATIMETHEREWASABOYCALLEDJACKHELIVEDWITHHISMOTHERTHEYWEREVERYPOORALLTHEYHADWASACOWONEMORNINGJACKSMOTHERTOLDJACKTOTAKETHEIRCOWTOMARKETANDSELLHERONTHEWAYJACKMETAMANHEGAVEJACKSOMEMAGICBEANSFORTHECOWJACKTOOKTHEBEANSANDWENTBACKHOMEWHENJACKSMOTHERSAWTHEBEANSSHEWASVERYANGRYSHETHREWTHEBEANSOUTOFTHEWINDOWTHENEXTMORNINGJACKLOOKEDOUTOFTHEWINDOWTHEREWASAGIANTBEANSTALKHEWENTOUTSIDEANDSTARTEDTOCLIMBTHEBEANSTALKHECLIMBEDUPTOTHESKYTHROUGHTHECLOUDSJACKSAWABEAUTIFULCASTLEHEWENTINSIDEJACKHEARDAVOICEFEEFIFOFUMJACKRANINTOACUPBOARDANENORMOUSGIANTCAMEINTOTHEROOMANDSATDOWNONTHETABLETHEREWASAHENANDAGOLDENHARPLAYSAIDTHEGIANTTHEHENLAIDANEGGITWASMADEOFGOLDSINGSAIDTHEGIANTTHEHARPBEGANTOSINGSOONTHEGIANTWASASLEEPJACKJUMPEDOUTOFTHECUPBOARDHETOOKTHEHENANDTHEHARPSUDDENLYTHEHARPSANGHELPMASTERTHEGIANTWOKEUPANDSHOUTEDFEEFIFOFUMJACKRANANDSTARTEDCLIMBINGDOWNTHEBEANSTALKTHEGIANTCAMEDOWNAFTERHIMJACKSHOUTEDMOTHERHELPJACKSMOTHERTOOKANAXEANDCHOPPEDDOWNTHEBEANSTALKTHEGIANTFELLANDCRASHEDTOTHEGROUNDNOBODYEVERSAWHIMAGAINWITHTHEGOLDENEGGSANDTHEMAGICHARPJACKANDHISMOTHERLIVEDHAPPILYEVERAFTER

1. **Documentation of the iterations to get to the plaintext message.**

(20 points)

* Overall there were about 24 iterations, but there was much trial and error  
  # no duplicate letters
* # V == C
* # M == T ##
* # B == U
* # K == N
* # F == S
* # D == O
* # E == Q
* # L == Z
* # R == W
* # P == K
* # A == H
* # X == D
* # I == A
* # Q == J
* # J == F
* # Z == R
* # C == M
* # S == I
* # U == P ##
* # N == V
* # O == B
* # Y == E
* # T == X
* # W == L
* # G == Y
* # H == G

1. **Beside the English language frequency of each character what else could you have calculated to help you find the plaintext?**

(10 points)

* 1. Explain the difference between the sliding window method and the block method.

(10 points)

* The sliding window method allows you to locate a section of letters as you slide through the text to identify patterns. It's usually used for simple substitutions and is a lot of manual work, but can be quite effective in decrypting ciphertext. The block method however is usually in big chunks and uses xor to find and locate words and decrypt sections of data and is used in modern algorithms.

1. **Explain the difference between conducting an exhaustive key search vs. English language character frequency and explain how your results supported or disproved the number of attempts needed to correctly decrypt the message.**

(10 points)

* An exhaustive key search does every single combination of letters in the cipher text and is usually 26! in size leading to a ridiculously long combination of iterations until you find what you're looking for. However, it is a way to brute force the key. English language character frequency allows you to combine the most frequent characters in the English language to be able to find words located within the ciphertext and slowly work your way up to an actual full sentence.

1. **Find a character frequency distribution for another language. Provide the frequency distribution in your lab report \*not a link, but the actual distribution) and be sure to identify which language it is.**

(10 points)

* French
  + E -> 15.10
  + A -> 8.13
  + S -> 7.91
  + T -> 7.11
  + I -> 6.94
  + R -> 6.94
  + N -> 6.42
  + U -> 6.05
  + L -> 5.68
  + O -> 5.27
  + D -> 3.23
  + M -> 3.23
  + C -> 3.15
  + P -> 3.03
  + É -> 2.13
  + V -> 1.83
  + H -> 1.08
  + G -> .97
  + F -> .96
  + B -> .93
  + Q -> .89
  + J -> .71
  + À ->.54
  + X -> .42
  + È -> .35
  + Ê -> .24
  + Z -> .21
  + Y ->.19
  + K -> .16
  + Ô -> .07
  + Û -> .05
  + W -> .04
  + Â -> .03
  + Î -> .03
  + Ü -> .02
  + Ù ->.02
  + Ë -> .01
  + Œ ->.01
  + Ç -> <.01
  + Ï -> <.01