**CText-1.txt**

The cipher text from CText-1.txt is encrypted using monoalphabetic cipher.

* The first step that is used in decrypting CText1 is by finding which letter has the most frequency by using the function that shows the frequency distribution graph of letters in Krypto.

A black background with white dots

AI-generated content may be incorrect.

* From the above diagram it can be seen that ‘R’ is the most frequently seen character in the text file.
* By using the diagram, ‘R’ character is mapped to ‘E’ since E is the most frequent character in the alphabet. ‘T’, ‘A’, ‘O’, ‘I’, ‘N’, ‘S’, ‘H’ and ‘R’ are mapped according to what characters appear frequently after ‘R’.
* After substituting the characters using Krypto, the following characters are used to decrypt the cipher text.

A screenshot of a computer

AI-generated content may be incorrect.

The following is the decrypted string using the above keys for CText-1

A screenshot of a computer

AI-generated content may be incorrect.

**CText-2.txt**

The cipher text from CText-2.txt is encrypted using Vigenère cipher.

* The first step that is used to decrypt the cipher text is to find the length of the key. To find the length of the key, an index of coincidence is needed. Index of coincidence is found by using the function to calculate the index of coincidence from Krypto.

A screenshot of a computer

AI-generated content may be incorrect.

* By using the above function, we take index of coincidence as 7 since it is the closest index of coincidence as English alphabet.
* The ciphertext is split into columns based on key length and count the frequencies of letter in each group.
* Count the shift by assuming the most frequent characters ‘E’, ‘T’ and other common letters.
* After shifting, we get QAYEQII as the encrypted key. After decrypting, we get VAMPIRE as the key for decryption.

The following is the decrypted plain text for CText-2

A black screen with white text

AI-generated content may be incorrect.