EXTRA CREDIT I

Use frequency analysis to break Caesar's cipher (this is another brute force method to break the cipher).

Frequency analysis relies on the fact that some letters (or combination of letters) occur more in a language, regardless of the text size. For example, in English the letters E, A are the most frequent, while the Z and Q are the least frequent; miscellaneous fact: see *Etaion shrdlu*[1]). The distribution of all the characters in English is depicted in the figure below:

E	T	A	O	I	N	S	H	R	D	L	U	C
12.7	9.1	8.2	7.5	7.0	6.7	6.3	6.1	6.0	4.3	4.0	2.8	2.8
M	**W**	**F**	**Y**	**G**	**P**	**B**	**V**	**K**	**X**	**J**	**Q**	**Z**
2.4	2.4	2.2	2.0	2.0	1.9	1.5	1.0	0.8	0.2	0.2	0.1	0.1

Figure 1. *Distribution letters in English (image source: https://ibmathsresources.com/tag/vigenere-cipher/)*

The idea of this method is to compare the frequency of the letters with the Chi-Squared distance:

$$\chi^2(C,E) = \sum_{i='a'}^{i='z'} \frac{(C_i - E_i)^2}{E_i}$$

, where C_i represents the occurrence of the i[th] character, and E_i is the expected count of the i[th] character of the alphabet.

The formula seems complicated at a first glance, but it is really not that complicated. Basically, for each possible character (*i* goes from 'a' to 'z'), we measure the discrepancy between how often it appeared in the encrypted text (C_i) and how often it is expected to appear in English texts (E_i); the difference $C_i - E_i$ is squared such that we remove negative signs. The division by E_i is simply a normalization factor.

The lower the Chi square distance $\chi^2(C,E)$, the more similar the histograms C and E are.
As this algorithm is also a brute force method to break the cipher, you should compute the histogram for all possible shifts, and compute the Chi Squared distance between these histograms and the average distribution of the characters in English. The shift with the lowest Chi Squared distance is the solution.

You can find more information about this algorithm here:
https://ibmathsresources.com/2014/06/15/using-chi-squared-to-crack-codes/ .

To sum up, to solve this problem you need to:

[1] https://en.wikipedia.org/wiki/Etaoin_shrdlu

- Write a function that reads the distribution of the letters from a file (*distribution.txt*) and stores it into an array. The frequency of the letter 'a' is stored on the first line of the file (as a floating point number), the frequency of the letter 'b' is stored on the second line of the file etc.
- Write a function that computes the normalized frequency of each character (a histogram) in a text.
- Write a function that computes the Chi squared distance between two histograms.
- Write a function that breaks the Caesar's cipher using frequency analysis: iteratively shifts the encrypted code through all the possible permutations, computes the Chi squared distance between each permutation and the approximate distribution of letters in English, and returns the permutation with the least Chi squared distance as the solution.