



Cipher Challenge

During World War II, Alan Turing (who is considered the father of modern computing) used computational analysis, and created the first computer to decrypt German messages. Given a key, the computer could decrypt messages at an alarming pace.

You are provided a set of plain text files: `encrypted.txt`, `encrypted_hard.txt`, and `plain.txt`.

You are to provide a project that decrypts `encrypted.txt`, using `plain.txt` as a base. The `plain.txt` file contains the literary works of William Shakespeare. There are various language patterns to create a computational algorithm for various types of analysis.

Please note the following:

- You can use a 1 for 1 type of algorithm. Which means, an uppercase character is the same as a lowercase character.
- Punctuation is punctuation, there is no decryption needed on these characters.
- Spaces are spaces, there is no decryption needed on these either.
- You may use any language you are comfortable with.
- There are no extra symbols used for the cipher.
- You may keep or remove the Project Gutenberg text in `plain.txt`. It has very little impact on the result.

Please provide the following in your project:

- The decrypted output of `encrypted.txt` in a plain text file.
- A project that contains the code you used to encrypt the file (how you structure your project will have an impact on the challenge as well).
- Note: A brute force solution will not suffice. Take into account that, like the Germans in WWII, we would modify the key at a moment's notice.
- Please provide a solution that shows how you derived your cipher key.

Impress us with:

- A decrypted output of `encrypted_hard.txt`.

- A well documented project that provides various best practices.
- A well structured API that can be used to decrypt files using a similar algorithm.
- A well structured front-end experience.

Please note: that the above are not requirements, but will help us determine your level of experience with the language(s) and tooling you provided.