# A brief context

This code challenge is inspired in a real story. 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. He realized that all encrypted German messages started with "Heil Hitler" and used that as a seed to find the encryption algorithm.

# The Challenge

You are provided with an encrypted.txt file. It contains an encrypted text of William Shakepeare. It uses a 1 for 1 encryption. This means that each character has been replaced with another character. So, there is a substitution table where, for example, every "A" is replaced with an "X", every "B" is replaced with a "C", every "C" is replaced with a "G" and so on.

In this example the world ABACA will appear as XCXGX.

Some other details.

- **This is case-insensitive. A = a**

**- Punctuation is punctuation. No need to decrypt them.**

**- Spaces are spaces. No need to decrypt these.**

**- There are no extra symbols used for this cipher.**

You are to provide a project that decrypts encrypted.txt and writes the result in a output.txt file.

We also provide you with a plain.txt file that contains a Shakespeare text similar to the one used encrypted in encrypted.txt. You can use this file to derive language patterns that help you find your solution.

# Project

Create a project, in the language of your preference, that gets the encypred.txt file (and optionally the plain.txt file) and returns an output.txt file with the decrypted text.

Give us a ZIP with

* A brief explanation of your thought process and how you approach the solution. 10 lines
* How do you find the decryption table? 10 lines
* How do you test that the decryption table is the right one? 10 lines
* The project code including the documentation.
* Compiling and running instructions if needed.
* The output.txt, result of your algorithm.

Extra points for

* A solution for encrypted\_hard.txt file.
* 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.

There are not required by twill help us determine your level of experience with the language(s) and tools you used.