Assignment 1

Programming problem: For this assignment, you can choose to work with any programming language.

Provided input file:

- `sample-secret-message-caesar-ciphertext.txt`: This text file contains the ciphertexts of four (4) messages (in 4 separate lines) encrypted using the Caesar cipher that we discussed in class. Note that the messages might not be generated using the same shift amount. You can assume that the messages only use the English alphabet.
- `sample-solutions.txt`: This text file contains the plaintext of the four ciphertexts in the `sample-secret-message-caesar-ciphertext.txt` file, with the corresponding correct shift amounts.
- `en-us.dic`: this dictionary file contains a list of English words that you can use to check for the correct shift amount.

Problem description:

- For this problem, you need to implement a program to break the Caesar ciphers used to generate the ciphertexts from a given input file. An input file should have the same format as the `sample-secret-message-caesar-ciphertext.txt` file.
- The program should retrieve the original messages (in plaintext) and output them to an output file in the same format as the `sample-solutions.txt` file. Your program can use the brute-force attack approach to try with all possible shift amounts to generate the Caesar cipher secrets.

Requirements:

- The program should do the following:
    1. Prompt the user to ask for the input ciphertexts file.
    2. Read in the provided input file. For each message, perform the decryption to find the correct shift amounts.
       Note that your program should identify the correct shift amount automatically using the provided dictionary file `en-us.dic`.
    3. Print the plaintexts and correct shift amounts for each message to the console in the same order as their ciphertexts appear in the input file.
    4. Output the plaintexts and correct shift amounts for each message to an output file in the same order as their ciphertexts appear in the input file.
- Your program should be well-commented.
- You must cite any sources that you use to complete the assignment.

Grading and Submission Instructions:

- We will test the program with a different input file. **Please make sure to include a document with instructions on how to run the program and perform the test following the requirements in the previous section. The instructions should not depend on any use of an IDE.**
- Please submit the following:
    1. The source code of your program.
    2. A document with instructions on how to set up and run your program. This document should include instructions on how to retrieve the expected output file when testing your program with the input file.
- Please compress all required files and submit them as a zip file.

Grading Rubric:

Task	**Point**
Correct outputs (there are 10 messages).	70
The program can compile and run successfully.	20
Well-commented code and documentation.	10
Total	100