CSCE 44303/54203 Homework 1 (Programming)

Release date: August 24, 2024

Due date: August 30, 2024

Full Grade: 60 pts

I. Task Description

In this assignment, you need to implement the Enhanced Caesar Cipher over the English alphabet (see Module 2 slides).

Part 1: Implement both encryption and decryption. For encryption, given a plaintext message (i.e., a sentence that contains multiple English words separated by spaces) and a key (i.e., a positive integer), your program should be able generate its ciphertext. For decryption, given a ciphertext (i.e., multiple blocks of English letters separated by spaces between blocks where each block corresponds to an English word) and a key, your program should be able to decrypt it and get the plaintext.

Part 2: Implement a brute-force attack that can decipher any ciphertext encrypted using the Enhanced Caesar Cipher where the corresponding plaintext consists of English words from a certain vocabulary specified in a text file. The plaintext message is not necessarily a sentence from the file; it could consist of words from multiple sentences. Specifically, given a particular ciphertext and the vocabulary text file, your program should be able to find the key *n* and the plaintext. For your convenience, a sample vocabulary text file *sample.txt* is attached. Note that your algorithm should work for other vocabulary file as well.

II. Tests

You need to demo your program to the grader or instructor. A demo sign-up sheet will be distributed in class later. During the demo, your program will be tested in the following ways.

Test of Part 1: For the encryption function, your program needs to take a manually input plaintext message (i.e., a sentence that contains multiple English words separated by spaces) and a key (i.e., a positive integer) from the command line user interface, and output the ciphertext to the command line. As an example:

- User-input message: hello world
- User-input key: 1
- Output: ifmmp xpsme

For the decryption function, your program needs to take a manually input ciphertext message (i.e., multiple blocks of English letters separated by spaces between blocks where each block corresponds to an English word) and a key (i.e., a positive integer) from the command line user interface, and output the plaintext to the command line. As an example:

- User-input ciphertext message: ifmmp xpsme
- User-input key: 1
- Output: hello world

Test of Part 2: Your program needs to take a manually input ciphertext message (i.e., multiple blocks of English letters separated by spaces between blocks) and a manually specified vocabulary text file name from the command line user interface, and output the deciphered plaintext message and the recovered encryption key to the command line interface. The test ciphertext will be chosen such that all the corresponding plaintext words are from the vocabulary file (but not necessarily from a sentence). As an example:

- User-input ciphertext message: jo dpnqvufs xpsme
- User-specified vocabulary text file: sample.txt
- Output:
    - Key: 1
    - Plaintext Message: in computer world

III. Other Instructions

Any programming language is fine.

Submit your code to Blackboard as a .zip file named in this format: HW1.YourLastName.YourFirstName.zip.