Caesar Cipher Encryption/Decryption Program

Purpose: Encrypt or decrypt basic text files using the Caesar cipher and a key

System requirements: command line capable of running C++ compiled programs

Performance requirements: less than a second for short text files (~100 words), should be capable of longer text files

Input: Standard txt file. The program can handle both uppercase and lowercase letters standard to the English alphabet (A-Z and a-z). Spaces, numbers, and special characters will be ignored during the encryption/decryption process, and will be output directly into the output file.

Functional requirements: Upon starting the program, the user will be asked if they would like to perform encryption or decryption, and will enter 1 or 0, respectively, and press enter. The program will not accept numbers except for 0 or 1, and it is not designed to handle non-numerical characters. The user will then be asked for an input file name. It is suggested that the .txt file will be in the same folder as the program for ease of use. If the input file is not found, or the name is entered incorrectly, the program will terminate. Once an input file is accepted, the program will ask for an output file name. The user can choose an existing file (which will be overwritten), or the program will create a file with the chosen name. Lastly, the program will ask for the user to input a key, which can be an uppercase or lowercase alphabetic letter. Choosing the letter A or a will result in no encryption/decryption. If a non-alphabetic character is entered, the user will be prompted to enter a new key. Once the key is entered, the program will generate the output text file and terminate. Case will be preserved. Check the output text file for the encrypted or decrypted file.

Output: standard text file with encrypted/decrypted text.

Part 4: Caesar Decryption with English Detection Program

Purpose: Crack an encrypted text file using the Caesar cipher and output the key as well as English text

System requirements: command line capable of running C++ compiled programs

Performance requirements: l**ess than a second for short text files (~100 words), should be capable of longer text files**

Input: Standard txt file. The program can handle both uppercase and lowercase letters standard to the English alphabet (A-Z and a-z). Spaces, numbers, and special characters will be ignored during the decryption process and carried over to the output file.

Functional requirements: Upon starting the program, the user will be asked asked for an input file name. It is suggested that the .txt file will be in the same folder as the program for ease of use. If the input file is not found, or the name is entered incorrectly, the program will terminate. Case will be preserved. Check the output text file for the decrypted file.

Methodology: The program will use a frequency analysis scorecard to determine which key (of the Caesar cipher) produces English text. Using a table of the most common letters in the English language, each letter is ranked by frequency. For example, E is the most common letter in the English language, T is the second most common, and the least common letters are Z, J, Q. After checking all 26 possible keys, the program will output the key of the highest score, and print the highest scoring decrypted string.

Output: console text with English decrypted text