Vigenère Cipher Documentation

Course
CIS-7

Section
27646

Date
December 10, 2023

Team
DC

Author
Daniel Chvat

1 Project Purpose

The Vigenère Cipher consists of a string and a key. Each character in the string is shifted based on the corresponding key character to get an Encrypted Character. The same process can be done backwards using the Encrypted Character and the same key to get the Decrypted/Original character back. It is possible to do the Vigenère Cipher by hand however for long strings it can get quite tiresome. This program aims to quicken the process by having the computer do all the work for the user using discrete structures concepts in the C++ programming language.

2 User Interface

The program interacts with the user by first prompting them to enter either E or D for Encryption Mode and Decryption Mode respectively. The user is then prompted to enter text to be encrypted or decrypted depending on which mode they are in. Then the user is asked to enter the key to encrypt or decrypt the text with. The program then runs the encryption or decryption function and displays the resulting text to the user.

3 Solution for Encryption

Characters between A – Z are translated into numerical values ranging from 0 to 25. If we let E_i denote the encrypted character, P_i denotes the plaintext character, and K_i denote the corresponding key character we find E_i using the following formulas:

$$E_i = \text{'A'} + [(P_i - \text{'A'}) + (K_i - \text{'A'})] \bmod 26 \text{ (Uppercase)}$$
$$E_i = \text{'a'} + [(P_i - \text{'a'}) + (K_i - \text{'a'})] \bmod 26 \text{ (Lowercase)}$$

4 Solution for Decryption

Characters between A – Z are translated into numerical values ranging from 0 to 25. If we let D_i denote the Decrypted character, E_i denote the Encrypted character, and K_i denote the corresponding key character we find D_i using the following formulas:

$$E_i = \text{'A'} + [(P_i - \text{'A'}) - (K_i - \text{'A'}) + 26] \bmod 26 \text{ (Uppercase)}$$
$$E_i = \text{'a'} + [(P_i - \text{'a'}) - (K_i - \text{'a'}) + 26] \bmod 26 \text{ (Lowercase)}$$

4 Program Limitations

- The program can only encrypt or decrypt characters ranging between A – Z.
- The Capitalization of the key character and plaintext string character must match each other, or results will be incorrect.
- The program can only encrypt, or decrypt characters stored in the ascii code other encoding formats are not supported.
- Time Complexity O(n)

5 Discrete Structures Concepts Utilized

Concept	Function in Program	Location in Code (Line #)
Functions	Take in key and plaintext and output Encrypted or Decrypted text depending on which function is called	5, 17
Logic	Check if character is upper or lower case and encrypt/decrypt that letter appropriately	9 – 13, 21 - 25
Modulo	Keep character value within range 0 – 25 inclusive	10, 12, 22, 24

6 Improving Limitations

- Add support for custom ranges so that the user can input two characters denoting the start and end of the range.
- Add support for new encoding formats using UTF-16 for example and let the user choose which encoding format they want to use. This combined with custom ranges will make the program much more versatile and add support for other languages as well.
- Add presets for other languages which the user could select from the menu.
- Make all character's upper case or lower case so we don't have to deal with the key and plaintext not being the same capitalization.

7 Pseudocode

FUNCTION encrypt (text, key)
    CREATE a placeholder called EncryptedText
    FOR each character IN text
        Get corresponding key character
        IF text character is uppercase
            ADD key to text keeping within 'A' – 'Z'
            ADD result to EncryptedText
        ELSEIF text character is lowercase
            Add key to text keeping within 'a' – 'z'
            ADD result to EncryptedText
        ENDIF
    RETURN EncryptedText

FUNCTION decrypt (text, key)
    Create a placeholder called DecryptedText
    FOR each character IN text
        Get corresponding key character
        IF text character is uppercase
            SUBTRACT key from text PLUS 26 keeping within 'A' – 'Z'
            ADD result to EncryptedText
        ELSEIF text character is lowercase
            SUBTRACT key from text PLUS 26 keeping within 'a' – 'z'

```
                ADD result to DecryptedText
            ENDIF
        RETURN DecryptedText

FUNCTION main (nothing)
        CREATE placeholders called choice, original, key, encrypted and decrypted
        DO
            PRINT "Vigenère Cipher Program"
            PRINT NEWLINE
            PRINT "-----------------------"
            PRINT NEWLINE
            PRINT "Please type E for Encrypt | D for Decrypt | Anything Else to Exit: "
            INPUT into choice
            CONVERT choice to uppercase if not already uppercase
            IF choice is ‘E’
                PRINT "Text to be Encrypted: "
                INPUT into original
                PRINT "Key to Encrypt with: "
                INPUT into key
                CALL encrypt (original, key) and store result in encrypted
                PRINT “Encrypted Text: [value of encrypted placeholder]”
                PRINT NEWLINE
            ELSEIF choice is ‘D’
                PRINT "Text to be Decrypted: "
                INPUT into original
                PRINT "Key to Decrypt with: "
                INPUT into key
                CALL decrypt (original, key) and store result in decrypted
                PRINT “Decrypted Text: [value of decrypted placeholder]”
                PRINT NEWLINE
            ENDIF
        UNITL choice does NOT EQUAL ‘E’ OR choice does NOT EQUAL ‘D’

        RETURN 0
```

8 Git

Project Files along with various testcases and compiling instructions can be found under my git repository https://github.com/DanielChvat/CIS-7