CIS7 Project Documentation Guide

Team: Astro

Members: Anooj Patel and Bruce Palmer

https://github.com/Gravuun/Vigenere-Cipher

Goal: The goal of the project is to create a method to encrypt and decrypt messages for users. Using the Vigenere cipher the users are able to enter their message and a key to encrypt or decrypt a message. If so desired, while encrypting a random key may be generated and used.

Calculations: To facilitate the process the characters of the string are converted into numeric values corresponding to letter position in the alphabet minus one. Then the numeric value of the character in the key that shares the same position as the character in the message is added. To account for overflow the modulus of the sum is taken with respect to 26 (number of characters in the alphabet). That values are then translated back into its corresponding alphabetical value and added to the final string.

Discrete Structure implementation: The use of a standardized cipher algorithm in the form of the Vigenere cipher. Also, the usage of modulus deal with overflow.

Limitations: this program does not encrypt or decrypt non-alphabet characters in either the key or message. In the key not alphabetical characters are removed.

Recommendations for improvement: Find a way to manipulate the algorithm to have the same functionality for alphabet characters as it currently does (wrap-around de/encryption) and also apply a shift not non-alphabet characters that does not result in alphabet characters being presented.

Pseudo Code:

Start Program

Set alphabet array to “abcdefghijklmnopqrstuvwxyz”

Function Main()

Declare random key generator

Declare: message, key, choice, repeat, random

Declare Boolean for edited

Do-while the user wants to continue with program

Initialize edited to false

Initialize random to false no ‘N’

Display menu options to user

Get “message” from user

If choice is 99

Break from do-while and end program

Else

Continue program

If choice is 1 or 2

Send user down encryption path

Else if choice is 3

Send user down decryption path

If choice was 2

Set random to Y , and set choice to 1

Get “message” from user

If user goes down encryption path

If user choose random enter randomKeyGen function, passing in message length

RandomKeyGen()

Declare variable for key string, to hold the random key

While the keyLength is greater than the position of the loop

Set random alphabet characters to the key

End RanomKeyGen function and return the key string

Else get desired key from user

Enter the function for Encryption(), of message, passing in message string and key string

Declare variable for a string for the encryptedMessage

While message length has not been reached with the increment variable

Set encryptedMessage to the encrypted char value at increment position

If char is upper

Set offset to 65

Else

Set offset to 97

Add the offset letter to the encryptedMessage string

Display the encrypted message to user

Return back to Main Function()

If user choose random key

Display random key that was used to encrypt message

If user goes down decryption path

Get “key” from user

Enter the function for decryption(), of message, passing in message string and key string

Declare variable for a string for the decryptedMessage

While message length has not been reached with the increment variable

Set decryptedMessage to the decrypted char value at increment position

If char is upper

Set offset to 65

Else

Set offset to 97

Use the offset to set the position of the MinusKey for decryption

Add the minusKey letter to the decryptedMessage string

Display the decrypted message to user

Return back to Main Function()

Display message asking user if they would like to Re-Run the program

Do while until user gives a valid response for yes or no

If user says yes

Restart the program

Else

Display message for exiting the program

End Program

Flow Chart:









