**Montgomery College**

**CMSC 203**

**Assignment 3 Design**

Class: CMSC203 CRN 21754

 Program: Assignment 3 Design

Instructor: Grinberg

 Summary of Description: (Give a brief description for a Program)

 Due Date: October 3rd 2022

 Integrity Pledge: I pledge that I have completed the programming assignment independently.

 I have not copied the code from a student or any source.

Student: Majd Haddad

**Part 1: Pseudo Code:**

Turn in pseudo-code for each of the methods specified in CryptoManager.java.   Refer to the [**Pseudocode Guideline**](#PSGdline)on how to write Pseudocode.

Is String In Bounds:

FOR every charAt(i) of plainText where i increments by 1 every iteration

IF charAt(i) is greater than or equal to LOWER\_RANGE AND less than or equal to UPPER\_RANGE THEN

RETURN true

ELSE

RETURN false

Caesar Encryption:

SET the encrypted equal to be blank

FOR every element I that iterates by 1 in plainText.charAt(i)

SET character letter equal to the case of key plus the plainText at char i

WHILE letter is greater than UPPER\_RANGE

Letter is equal to letter minus RANGE

CONCATENATE every index of the plainText to the encrypted version

RETURN the encrypted

Caesar Decryption:

SET the decrypted to be empty

DECLARE character variable letter

FOR every element i that iterates by 1 in plainText.charAt(i)

SET letter equal to the character cast of encryptedText.charAt(i) minus key

WHILE letter is less than LOWER\_RANGE

SET letter equal to letter plus RANGE

CONCATENATE the elements of i to decrypted

RETURN the decrypted

Bellaso Encryption:

SET the encrypted to be an empty String

DECLARE char variable letter

DECLARE integer named indexOfAddedLetter

FOR every value in plainText.charAt(i) incremented by 1

SET indexOfAddedLetter equal to I modulus bellasoStr length

SET letter equal to the casted character of bellasoStr.charAt(indexOfAddedLetter) plus the plainText.charAt(i)

WHILE sum is greater than UPPER\_RANGE

SET letter equal to letter minus RANGE

CONCATENATE the set of encrypted chars together in encrypted

RETURN encrypted String

Bellaso Decryption:

**SET the decrypted to be an empty String**

**DECLARE integers named numOfChar and letterNum**

**FOR every character in encryptedText length**

**SET numOfChar equal to I modulus bellasoStr length**

**SET letterNum equal to encryptedText.charAt(i) minus bellasoStr.charAt(numOfChar)**

**WHILE letterNum is less than LOWER\_RANGE**

**letterNum is equal to letterNum plus RANGE**

CONCATENATE the set of decrypted chars together in decrypted

**RETURN decrypted String**

**UML Diagram:**

**Table

Description automatically generated**

**Part 2: Comprehensive Test Plan**

Turn in a Test Plan table. Test Plan should include:

* at least two tests for the Caesar Cipher
* at least two for the Bellaso Cipher.
* at least one string that will fail because it has characters outside the acceptable ones.

\

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input text  First 2 are Caesar  And the next 2 are Cipher and the last is an invalid | Input Key | Encrypted (Caesar) | Encrypted (Bella) | Decrypt (method1) | Decrypt (method2) |
| MAJD | 7 | THQK |  | MAJD |  |
| WAKE UP | 3 | ZDNH#XS |  | WAKE UP |  |
| DURANT AND CURRY | ELEVEN |  | I!WWS"%MSZ%QZ^W/ |  | DURANT AND CURRY |
| LAMAR JACKSON | EIGHT |  | QJTI&%SHK\_XXU |  | LAMAR JACKSON |
| username@gmail.com | HEY | FAIL | FAIL | username@gmail.com | FAIL |

**Make sure your tests cover all the possible scenarios.**

REQUIRED OUTPUT

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Input MAJD w/ key = 7 Screenshot CAESAREncryption:

Graphical user interface, text, application, email

Description automatically generated

Input MAJD w/ key = 7 Screenshot CAESARDECRYPTION:

Graphical user interface, text, application, email

Description automatically generated

Input WAKE UP w/ key = 3 Caesar Encryption SS:

Graphical user interface, text, application, email

Description automatically generated  
Input WAKE UP w/ key = 3 Caesar DECRYPTION SS:

Graphical user interface, text, application, email

Description automatically generated

Input DURANT AND CURRY key = ELEVEN BELLASO ENCRYPTION SS:

Graphical user interface, text, application, email

Description automatically generated

Input DURANT AND CURRY key = ELEVEN BELLASO DECRYPTION SS:

Graphical user interface, text, application, email

Description automatically generated

LAMAR JACKSON key = EIGHT BELLASO ENCRYPTION SS:

Graphical user interface, text, application, email

Description automatically generated

LAMAR JACKSON key = EIGHT BELLASO DECRYPTION SS:

Graphical user interface, text, application, email

Description automatically generated

FAIL Screenshot:

GITHUB Screenshot:

**Lessons Learned** <Provide answers to the questions listed above>**:**

Write about your Learning Experience, highlighting your lessons learned and learning experience from working on this project.

What have you learned? I learned a lot about cybersecutrity including how to encrypt and decrypt bellaso and Caesar. I also learned a lot about the modulus and how important it can be.

What did you struggle with?

I struggled a lot with the ballaso portion of having the key match the same length of the plainText. I kept thinking nested loops were the answer to that question, however; once I played around with things and used I % the key.length inside a normal for loop, that made everything so much easier.

What would you do differently on your next project?

Next time, I would better understand the concepts of the process of bellaso and Caesar as I forgot the steps and didn’t read the instructions carefully.

What parts of this assignment were you successful with, and what parts (if any) were you not successful with?

I was successful with the inbounds, Caesar encryption and Caesar decryption for the most part, I did struggle a little with the output not displaying properly but other than that, those went pretty smoothly.

Provide any additional resources/links/videos you used to while working on this assignment/project.