/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Class: CMSC203 CRN 46667

Program: Assignment # 3

Instructor: Dr.Grinberg

Description: A program that contains the methods to encrypt/decrypt a plain text

string using a Caesar Cipher and Bellaso approaches.

Due: 07/12/2020

I pledge that I have completed the programming assignment independently.

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

Anthony Liu

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Pseudocode:

**stringInBounds –**

LOOP through each character of the String

IF current character in the String is greater than UPPER\_BOUND or less than LOWER\_BOUND

Return false

ELSE

Return true

**encryptCaesar-**

DECLARE empty String called caesarEncrypt

LOOP through each character of input String

DECLARE and SET charEncrypt to current char + key

WHILE charEncrypt is greater than int UPPER\_BOUND

SUBTRACT RANGE from charEncrypt

CONVERT charEncrypt back to char

ADD charEncrypt to caesarEncrypt

RETURN caesarEncrypt

**encryptBellaso-**

DECLARE empty String called bellasoEncrypt

LOOP through each character of String input

CALCULATE current key character index by using modulo and key length (i % length)

DECLARE and SET charEncrypt to current character + current key character

WHILE charEncrypt is greater than int UPPER\_BOUND

SUBTRACT RANGE from charEncrypt

CONVERT charEncrypt to char

ADD charEncrypt to bellasoEncrypt

RETURN bellasoEncrypt

**decryptCaesar-**

DECLARE empty String called decryptedCaesar

LOOP through each character of input String

DECLARE and SET charDecrypt to current char - key

WHILE charDecrypt is less than int LOWER\_BOUND

ADD RANGE to charEncrypt

CONVERT charDecrypt to char

ADD charDecrypt to decryptedCaesar

RETURN decryptedCaesar

**decryptBellaso-**

DECLARE empty String called decryptedBellaso

LOOP through each character of String input

CALCULATE current key character index by using modulo and key length (i % length)

DECLARE and SET charDecrypt to current character - current key character

WHILE charDecrypt is less than int LOWER\_BOUND

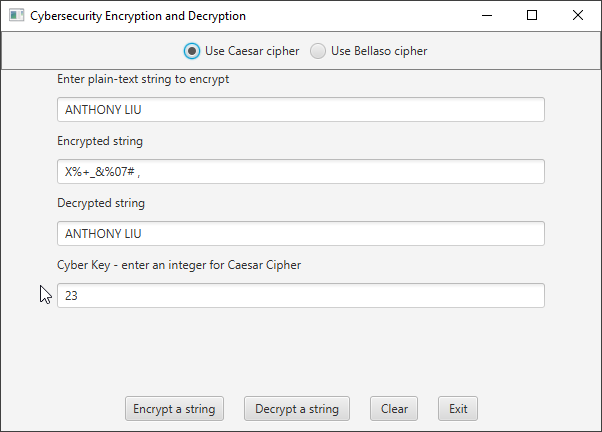
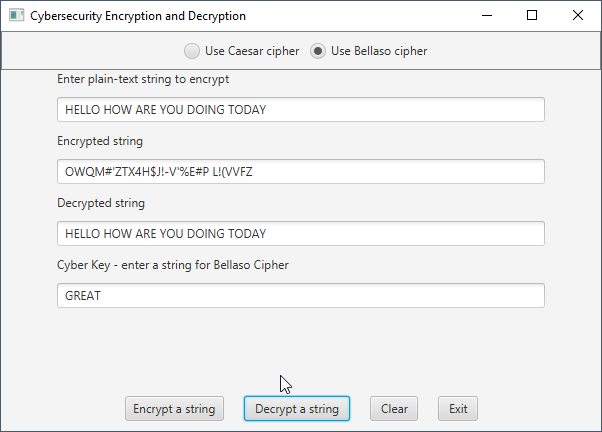
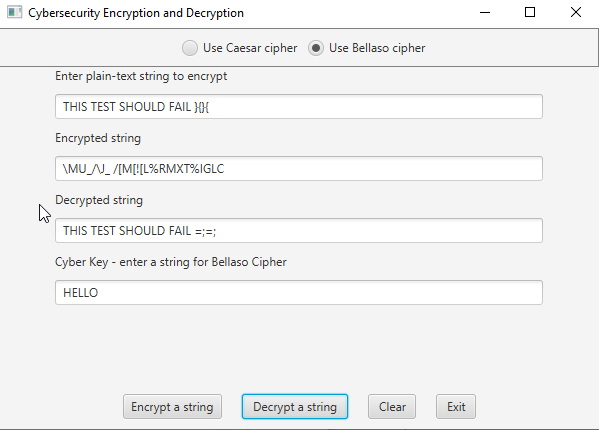
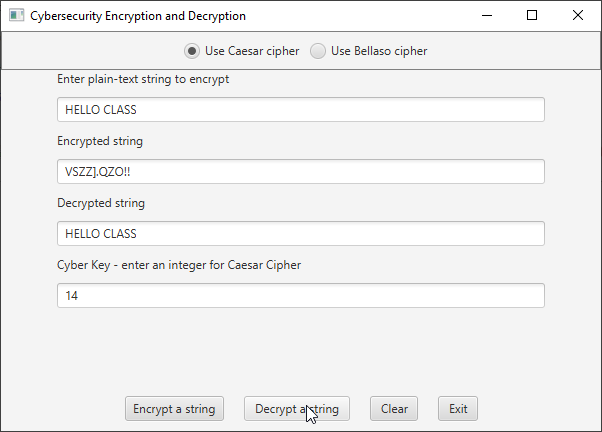
ADD RANGE to charDecrypt

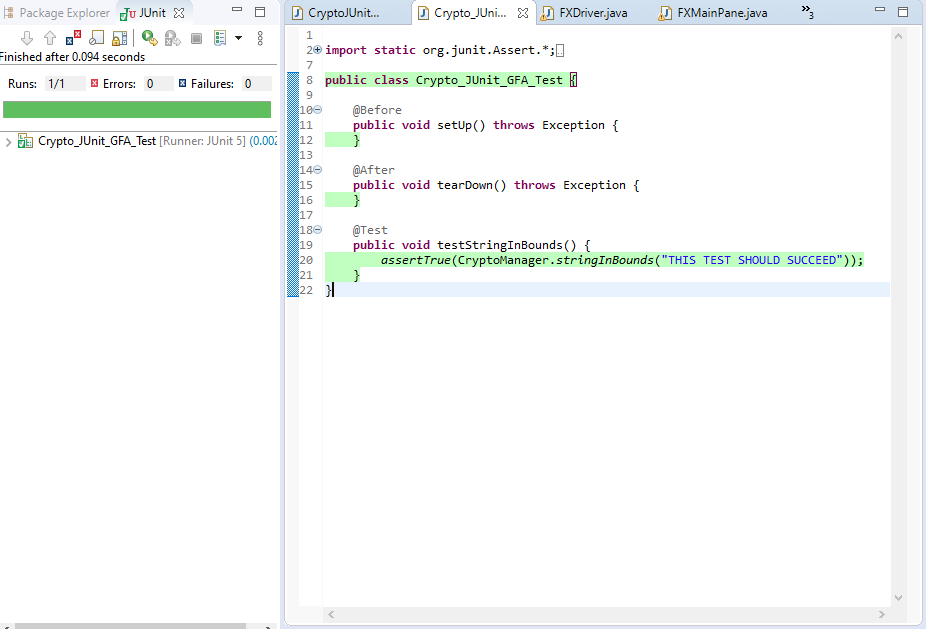
CONVERT charDecrypt to char

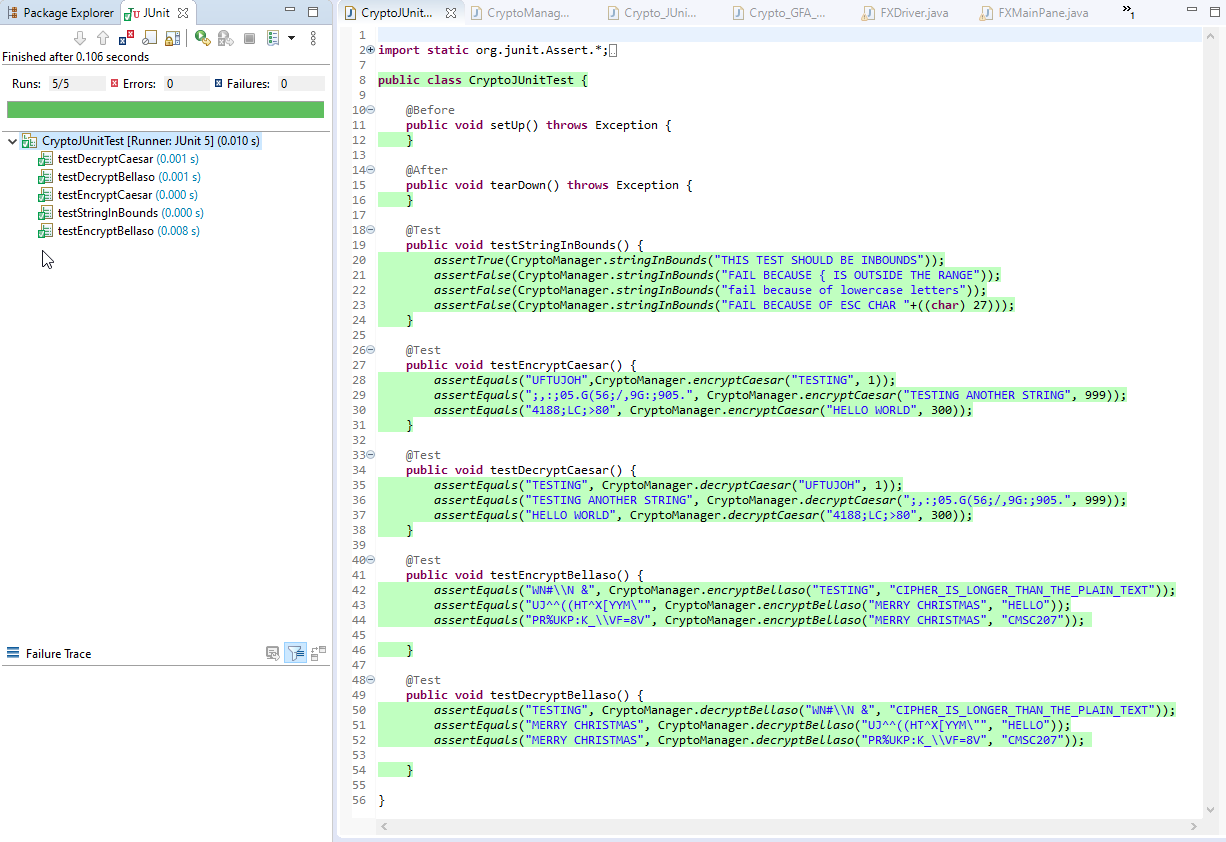
ADD charDecrypt to decryptedBellaso

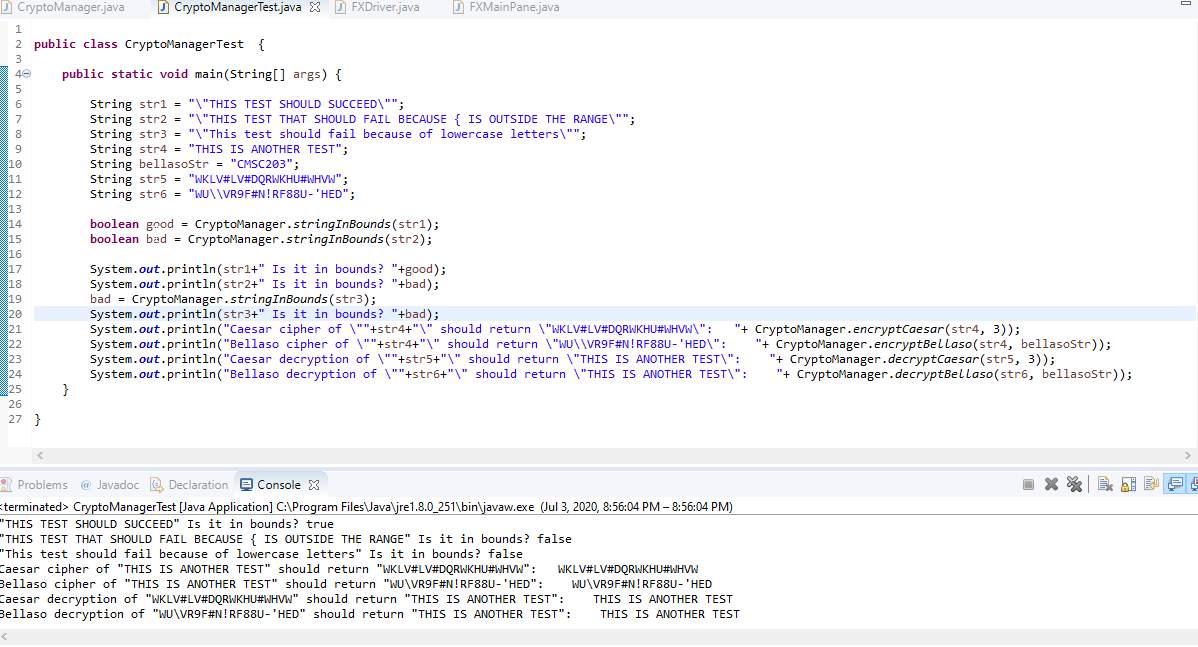
RETURN decryptedBellaso

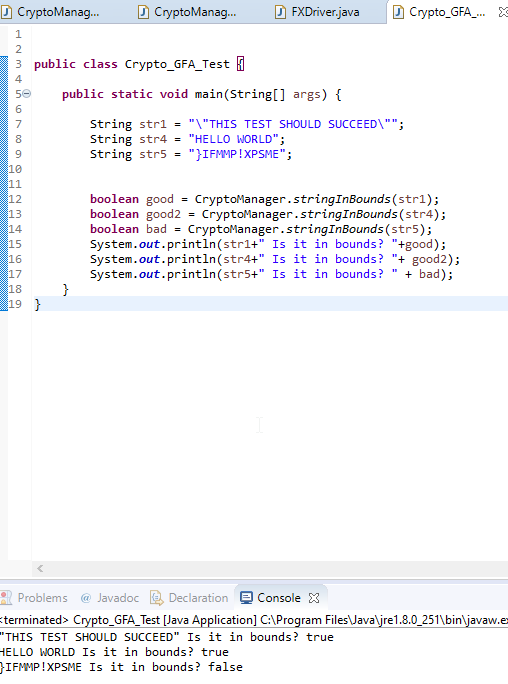
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input text | Input Key | Encrypted (method1) | Encrypted (method2) | Decrypt (method1) | Decrypt (method2) |
| ANTHONY LIU | 23 | X%+\_&%07# , |  | ANTHONY LIU |  |
| HELLO CLASS | 14 | VSZZ].QZO!! |  | HELLO CLASS |  |
| HELLO HOW ARE YOU DOING TODAY | GREAT |  | OWQM#'ZTX4H$J!-V'%E#P L!(VVFZ |  | HELLO HOW ARE YOU DOING TODAY |
| THIS TEST SHOULD FAIL }{}{ | HELLO |  | \MU\_/\J\_ /[M[![L%RMXT%IGLC |  | THIS TEST SHOULD FAIL =;=; |

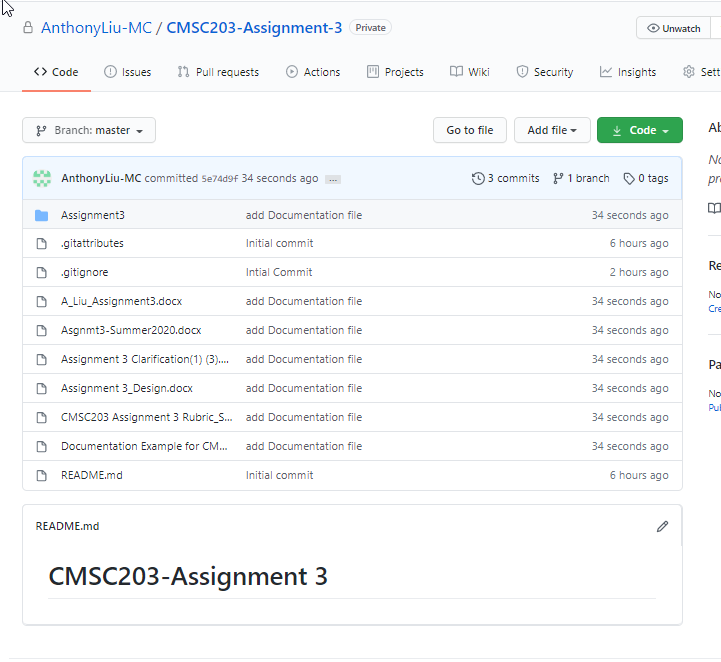
TEST CASES











Learning Experience

This assignment has shown to be the most difficult one we have had so far in this class. It tests us to think algorithmically to implement the encryption/decryption methods. I struggled initially with how to implement them but as soon as I got passed it, the rest of the lab was fairly straight forward. As with all programming, the task on hand is a lot easier to comprehend if you are able to break it down into small steps.

Assignment 3 Check List

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | **Assignment files:** |  |  |
|  | * FirstInitialLastName\_ Assignment#\_Moss.zip | **Yes** |  |
|  | * FirstInitialLastName\_Assignment#.docx/.pdf | **Yes** |  |
|  | * Source java files | **Yes** |  |
|  | **Program compiles** | **Yes** |  |
|  | **Program runs with desired outputs related to a Test Plan** | **Yes** |  |
|  | **Documentation file:** |  |  |
|  | * Comprehensive Test Plan | **Yes** |  |
|  | * Screenshots for each Test case listed in the Test Plan | **Yes** |  |
|  | * Screenshots of your GitHub account with submitted Assignment# (if required) | **Yes** |  |
|  | * UML Diagram (if required) | **N/A** |  |
|  | * Algorithms/Pseudocode (if required) | **Yes** |  |
|  | * Flowchart (if required) | **N/A** |  |
|  | * Lessons Learned | **Yes** |  |
|  | * Checklist is completed and included in the Documentation | **Yes** |  |