

Part 1: Pseudo Code:

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

Program start

Declare final char variable LOWER_BOUND as ' '

Declare final char variable UPPER_BOUND as '_'

Declare final integer variable RANGE as UPPER_BOUND - LOWER_BOUND + 1

Declare public static Boolean method stringInBounds with argument String `plainText`

 Declare Boolean variable `valid` as true

 FOR set integer `i` as 0, loops while `i` is less than the plaintext length, and increments `i` for each loop

 Declare char variable `c` as `plainText.charAt(i)`

 IF `c` is less than LOWER_BOUND or more than UPPER_BOUND

 SET `Valid` equal to false

 ELSE

 SET `Valid` equal to true

 Return `valid`

Declare public static String method encryptCaesar with arguments String `plaintext` and int `key`

 IF `stringInBounds` does not equal `plaintext`

 Return ""

declare `encryptC` as ""

FOR set integer `i` as 0, loops while `i` is less than the plaintext length, and increments `i` for each loop

 Declare `c` as `plainText.charAt(i)`

Declare `ec` equal to `c` converted to an integer value

 SET `ec` to add equal `key`

```
WHILE ec is greater than UPPER_BOUND
    SET ec to subtract equal RANGE
```

```
    SET encryptC add equals ec set to char value;
```

```
Return encryptC;
```

```
Declare encryptBellaso with arguments String plaintext and String bellasoStr
```

```
    Declare encryptB as ""
```

```
Declare be as bellasoStr.length()
```

```
    FOR set integer i as 0, loops while i is less than the plaintext length, and
    increments i for each loop
```

```
        Declare c as plainText.charAt(i)
```

```
Declare eb as (int)bellasoStr.charAt(be) added by (int)c
```

```
    WHILE eb is greater than UPPER_BOUND
```

```
        SET eb minus equal to RANGE
```

```
        Declare encryptB add equal to (char)eb
```

```
    Return encryptB
```

```
Declare decryptCaesar with arguments String encryptedText and int key
```

```
    Declares decryptC as ""
```

```
    FOR set integer i as 0, loops while i is less than the encryptedText length,
    and increments i for each loop
```

```
Declares c as encryptedText.charAt(i)
```

```
    Declares dc as (int)c
```

```
    SET subtract equal to key
```

```
WHILE dc is less than LOWER_BOUND
```

```
    SET dc subtract equal to RANGE
```

```
    SET decryptC as add equal to (char)dc
```

Return decryptC

Declare decryptBellaso with arguments String encryptedText and String bellasoStr

Declare decryptB as ""

Declare be as bellasoStr.length()

FOR set integer i as 0, loops while i is less than the plaintext length, and increments i for each loop

Declare c as encryptedText.charAt(i)

Declare db as (int)bellasoStr.charAt(be-i) subtracted by (int)c

WHILE db is less than the LOWER_BOUND

SET db add equal to RANGE

SET decryptB add equal to (char)db

Return decryptB

End Program

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	Input Key	Encrypted (Method 1)	Encrypted (Method 2)	Decrypt (Method 1)	Decrypt (Method 2)
HEY HOW ARE YOU	17	YV*1Y (1R#V1* &		HEY HOW ARE YOU	
OUTSTANDING	41	8>=<=*7-270		OUTSTANDING	

RUBBER BABY BUGGY BUMPERS	GREAT		Y'GCYY2GBV 2GV[N+%C)T"J S'		RUBBER BABY BUGGY BUMPERS
ITSY BITSY SPIDER	HELP		QY_)(GU\$!%_ QIQ"		ITSY BITSY SPIDER
failure	4rh				

Cybersecurity Encryption and Decryption

☒ Use Caesar cipher
☐ Use Bellaso cipher

Enter plain-text string to encrypt

HEY HOW ARE YOU

Encrypted string

YV*1Y (1R#V1* &

Decrypted string

HEY HOW ARE YOU

Cyber Key - enter an integer for Caesar Cipher

17

Encrypt a string

Decrypt a string

Clear

Exit

Cybersecurity Encryption and Decryption

☒ Use Caesar cipher ☐ Use Bellaso cipher

Enter plain-text string to encrypt

OUTSTANDING

Encrypted string

8>=<=*7-270

Decrypted string

OUTSTANDING

Cyber Key - enter an integer for Caesar Cipher

41

Encrypt a string Decrypt a string Clear Exit

Cybersecurity Encryption and Decryption

☐ Use Caesar cipher ☒ Use Bellaso cipher

Enter plain-text string to encrypt

RUBBER BABY BUGGY BUMPERS

Encrypted string

Y'GCYY2GBV 2GV[N+%C)T"JS'

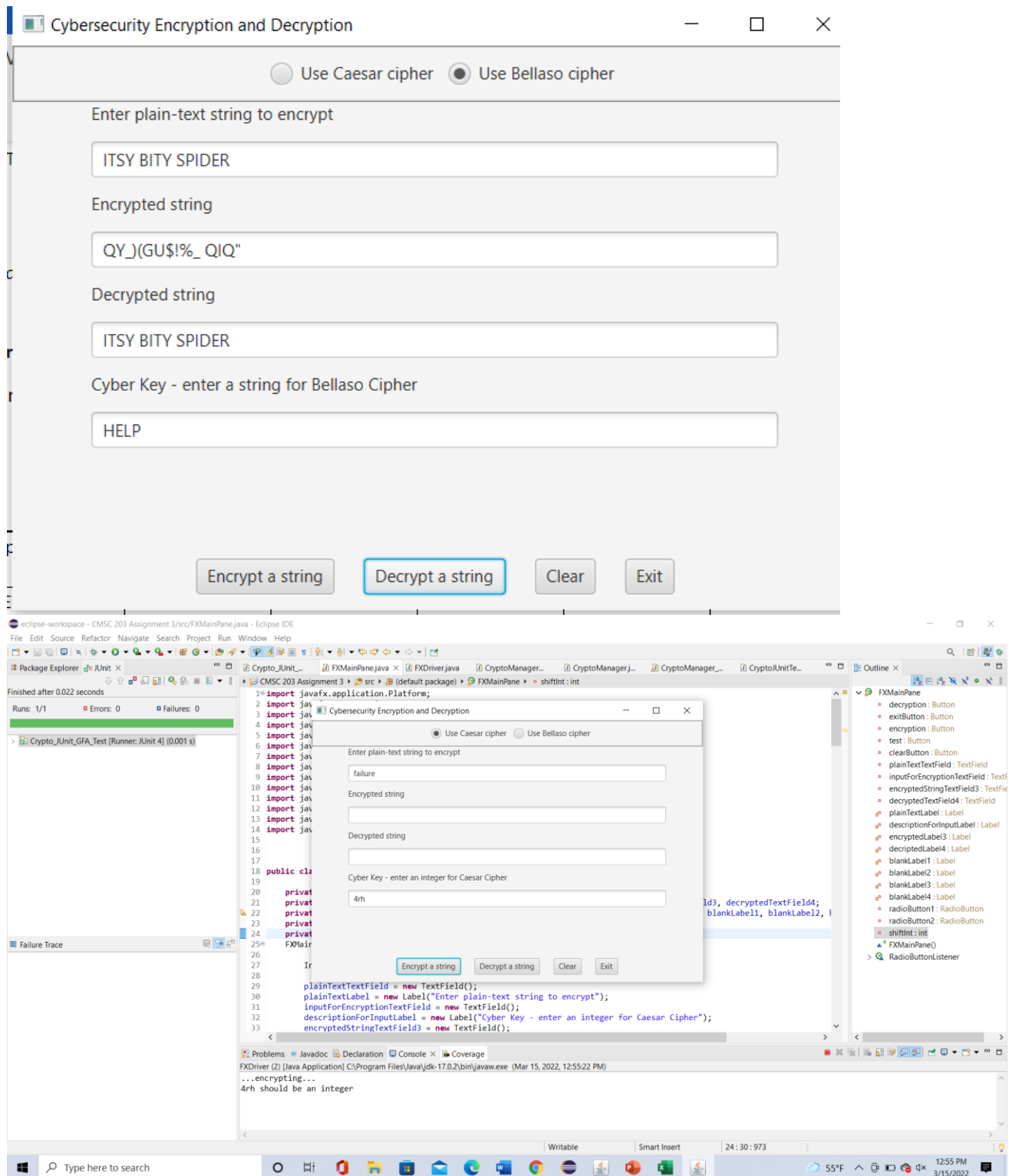
Decrypted string

RUBBER BABY BUGGY BUMPERS

Cyber Key - enter a string for Bellaso Cipher

GREAT

Encrypt a string Decrypt a string Clear Exit



Lessons Learned:

What have you learned? **I learned how to successfully encrypt and decrypt text**

What did you struggle with? **I struggled with the JUnit errors, the running the FX files, and the bellaso cipher**

What will you do differently on your next project? **I would look to the lab assignments and YouTube videos for more assistance on how to successfully finish an assignment.**

Include what parts of the project you were successful at, and what parts (if any) you were not successful at. **At first I had a hard time getting the box to appear, but after I added a VM argument, the box appeared and functioned as desired.**

Assignment 3 Check List (Fill out a column Y/N)

#		Y/N	Comments
	Assignment files:		
	• FirstInitialLastName_Assignment3_Moss.zip	Y	Original java files
	• FirstInitialLastName_Assignment3_Complete.zip	Y	Copied java files
	• FirstInitialLastName_Assignment#.docx/.pdf	Y	
	• Source java files	Y	
	Program compiles	Y	
	Program runs with desired outputs related to a Test Plan	Y	
	Documentation file:		
	• Comprehensive Test Plan	Y	
	• Screenshots for each Test case listed in the Test Plan	Y	
	• Screenshots of your GitHub account with submitted Assignment# (if required)	Y	
	• Algorithms/Pseudocode (if required)	Y	
	• Lessons Learned	Y	

	<ul style="list-style-type: none">• Checklist is completed and included in the Documentation	Y	
--	--	---	--